

Actas de las XIX Jenui. Castellón, 10-12 de julio 2013
ISBN: 978-84-695-8051-6 DOI: 10.6035/e-TiIT.2013.13
Páginas: 327-334

Entorno virtualizado de aprendizaje para facilitar el desarrollo de destrezas de programación

Germán Moltó, Oscar Sapena
Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València
46022 Valencia
{gmolto, osapena}@dsic.upv.es

Resumen

Esta comunicación resume la experiencia de creación y uso de un entorno virtualizado de aprendizaje de la programación para los alumnos de primer curso del Grado en Informática de la Universitat Politècnica de València durante el curso 2012/2013. Dicho entorno ofrece una configuración similar a la de los laboratorios de prácticas e integra un Corrector Automático de Programas así como vídeo-ejercicios de programación para crear un entorno virtual de aprendizaje interactivo. Además, puede ser ejecutado en los equipos de los alumnos sin necesidad de acceso a Internet y sin requerir costosas modificaciones de la instalación existente en los propios equipos. Esto permite a los alumnos recién llegados evitar la instalación del Sistema Operativo y las diferentes librerías y herramientas que se emplean durante dichas asignaturas. Se describe el proceso de creación y distribución del entorno (en la forma de una máquina virtual), así como la integración con vídeo-ejercicios didácticos y un corrector automático de programas. Finalmente se abordan las estadísticas de uso y los primeros resultados de la evaluación de la utilidad del mismo realizada por los estudiantes.

Abstract

This paper summarises the development and usage of a virtualised learning environment for the students of programming subjects in the Degree of Computer Science at the Universitat Politècnica de València during the academic course 2012/2013. Such environment exposes a configuration similar to the one used in the labs and integrates an Automatic Program Testing tool as well as video-exercises to create a unique interactive virtual learning environment. Besides, it can be executed in the student's computer without requiring Internet access and without major modifications in the student's computer. This enables freshmen to avoid dealing with the intricacies of installing the Opera-

ting System, libraries and tools recommended for those subjects. The process of creating and distributing this environment (in the shape of a virtual machine) is described, including the integration with learning videos and an automatic grading software. Finally, usage stats are covered as well as the first evaluation results of the usefulness of this service for the students.

Palabras clave

Programación, Virtualización, Entorno de Aprendizaje, Aprendizaje Autónomo

1. Introducción

El primer curso del Grado en Informática en la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València incluye dos asignaturas cuatrimestrales de programación, Introducción a la Informática y a la Programación (IIP) y Programación (PRG). En ellas se inicia al alumno en la construcción de programas, utilizando Java como lenguaje de programación. Además, se utiliza GNU/Linux como Sistema Operativo (SO) en los laboratorios de prácticas. Esto permite que los alumnos se introduzcan a usar este SO ya que gran parte de los alumnos no lo utiliza de forma habitual. En efecto, al inicio de la asignatura IIP (primer cuatrimestre de primer curso) se pasó una encuesta online con el objetivo de caracterizar los equipos de los alumnos y sus preferencias de uso de SO. La encuesta fue contestada por exactamente 100 alumnos y arrojó los siguientes resultados.

- **Equipo** Un 41 % disponen de un equipo de sobremesa mientras que un 59 % tiene un equipo portátil.
- **SO** Un 95 % utiliza de forma habitual Windows, un 2 % OS X y un 6 % GNU/Linux (pregunta con multirespuesta).
- **Memoria RAM** Un 12 % dispone de menos de

512 MBytes de RAM, un 16 % entre 1 y 2 GBytes, un 24 % entre 2 y 4 GBytes y un 48 % más de 4 GBytes.

De los resultados de la encuesta se deduce que la mayoría de los alumnos utiliza Windows sobre equipos generalmente portátiles y con más de 2 GBytes de RAM. Los alumnos disponen de laboratorios para realizar las prácticas donde utilizan equipos basados en GNU/Linux (en particular CentOS), junto con el resto de aplicaciones necesarias para la asignatura (Java Development Kit y el entorno de desarrollo BlueJ [1]). Sin embargo los laboratorios tienen unos determinados horarios de acceso que obligan a los alumnos a adaptarse a ellos para poder terminar ciertas prácticas o simplemente disponer del entorno de trabajo recomendado para realizar las actividades de clase.

Para paliar este problema, el departamento responsable de estas asignaturas, ofrece un servicio de Escritorios Virtuales¹ que permite acceder a un escritorio remoto GNU/Linux con el mismo entorno disponible en los laboratorios de prácticas. Para ello, es necesario un navegador web con soporte para Java e instalar un plugin de la tecnología NX (NoMachine)². Sin embargo, esta opción requiere de una conexión a Internet de banda ancha y con baja latencia para tener unos tiempos de respuesta de interacción con el escritorio apropiados. Aunque esta opción puede ser útil para conectarse de forma esporádica, algunos alumnos no disponen de conexión a Internet en periodos de vacaciones, cuando regresan a sus lugares de procedencia y, por tanto, no pueden aprovechar esta alternativa.

Con el auge en las técnicas de virtualización y los avances en el desarrollo de hipervisores, en la actualidad resulta sencillo crear una máquina virtual para que se ejecute por encima del SO del usuario. Esto permite al alumno ejecutar GNU/Linux dentro de una máquina virtual que a su vez se ejecuta sobre el SO del equipo del estudiante (típicamente Windows, como se ha visto anteriormente). De esta manera, el estudiante que quiere utilizar GNU/Linux en su equipo no precisa de formatear y reparticionar su equipo con la complejidad técnica asociada al proceso. Además, la máquina virtual puede incluir la instalación de aplicaciones específicas. Esto permite crear lo que en este artículo se define como Entorno Virtualizado para el Aprendizaje de la Programación (EVAP), una máquina virtual que incluye todo el material y las aplicaciones necesarias para dar soporte a la realización de las actividades de la asignatura.

En este artículo se detalla el proceso de creación y utilización de EVAP durante el curso 2012/2013. Tras la introducción, el resto del artículo está estructurado como sigue. En primer lugar, la sección 2 detalla los

trabajos relacionados en este campo. A continuación, la sección 3 discute la metodología empleada y detalla las herramientas y consideraciones involucradas en el desarrollo de EVAP. Posteriormente, la sección 4 describe las características avanzadas así como la integración de otras herramientas educativas en dicho entorno. Luego, la sección 5 comenta los principales resultados y discute las ventajas e inconvenientes de la aproximación propuesta. Finalmente, la sección 6 resume el artículo y plantea las líneas de trabajo futuras.

2. Trabajos relacionados

En 2006, cuando las técnicas de virtualización comenzaban a popularizarse, ya algunos profesores como Gómez Martín [3] abogaban por las ventajas de introducir máquinas virtuales en el contexto docente, especialmente para aquellas asignaturas en las que fuera necesario que el alumno tuviese privilegios de superusuario (*root*). En este artículo se pone como ejemplo asignaturas como Sistemas Operativos, que precisan por parte del alumno control completo sobre el equipo (en este caso sobre la máquina virtual) para realizar el proceso de instalación y configuración del mismo, sin afectar por tanto a las instalaciones de software existentes en los laboratorios de prácticas. En esta línea se encuentra el trabajo de Pardo [6] sobre redes de ordenadores, que utiliza máquinas virtuales clonadas a partir de una máquina virtual base con el objetivo de configurar múltiples redes entre las máquinas virtuales y afianzar los conceptos propuestos en la asignatura.

Sin embargo, el uso de máquinas virtuales para ofrecer a los alumnos entornos de trabajo virtualizados y auto-contenidos para una asignatura concreta es relativamente reciente en la comunidad académica (al menos su difusión pública). En un trabajo realizado en la Universidad de Washington³ se describe el uso de una máquina virtual basada en GNU/Linux con el objetivo de que el estudiante pueda utilizar este SO en su equipo, independientemente de su SO actual. La documentación no detalla más información sobre el software instalado en la misma, ni parece que su configuración esté encaminada a dar soporte una asignatura concreta. Por el contrario, en el trabajo de Malan⁴ se detalla el uso de una máquina virtual preconfigurada puesta a disposición de los alumnos para llevar a cabo las actividades propuestas en la asignatura CS50 de Harvard University, una asignatura de programación que aborda los lenguajes C, PHP, JavaScript, SQL, CSS y HTML. No se detallan las aplicaciones específicas instaladas en la máquina virtual, pero un aspecto interesante es que ha sido configurada para realizar copias periódicamente.

¹<http://evir.dsic.upv.es/>

²<http://www.nomachine.com/>

³<https://www.cs.washington.edu/lab/software/homeVMs>

⁴<https://manual.cs50.net/appliance>

cas de los ficheros en el directorio del usuario (a otro directorio) cada 10 minutos. Esto posibilita recuperar ficheros de código que el estudiante haya podido borrar de forma accidental durante el uso de la máquina virtual.

También existen iniciativas de estas características en España. Por ejemplo, en [6] se recomienda la instalación de una máquina virtual previamente creada por los docentes para dar soporte a la asignatura Arquitectura de Sistemas, impartida en la Universidad Carlos III de Madrid (UC3M). Esta máquina virtual se orienta a utilizar el compilador *gcc*, el depurador de programas *gdb* y el depurador de memoria *Valgrind*.

En nuestro trabajo se aborda la construcción de un entorno virtualizado de trabajo para facilitar la enseñanza de la programación en Java, construido como soporte para las asignaturas IIP y PRG, pero generalizable y reutilizable para otras asignaturas afines. A diferencia de otros trabajos, el entorno propuesto en este artículo no se limita a una simple máquina virtual con unas aplicaciones pre-instaladas. También integra dos herramientas ya utilizadas en las asignaturas: vídeo-ejercicios didácticos para el aprendizaje autónomo y un Corrector Automático de Programas para la evaluación automática y guiada de programas de los alumnos. La integración de ambas tecnologías en EVAP permite la construcción de un entorno virtual interactivo y desatendido para fomentar en el alumno el aprendizaje autónomo de destrezas de programación.

3. Metodología

Esta sección resume el proceso de construcción de EVAP (Entorno Virtualizado para el Aprendizaje de la Programación). En primer lugar se procede a describir brevemente la asignatura IIP sobre la que inicialmente se ha realizado la experiencia. Después, por motivos de completitud, se resume brevemente las técnicas de virtualización empleadas. A continuación se detallan las características básicas soportadas por EVAP. Luego, se describe la funcionalidad avanzada, mediante la integración de vídeo-ejercicios y el Corrector Automático de Programas. Finalmente, se aborda el proceso de distribución y recopilación de sugerencias de mejora.

3.1. La asignatura IIP

La asignatura IIP (Introducción a la Informática y a la Programación) es una asignatura de 6 créditos ECTS que se imparte en el primer cuatrimestre de primer curso del Grado en Ingeniería Informática de la Universitat Politècnica de València. La asignatura pretende que el alumno sea capaz de escribir programas correctos de baja complejidad en un lenguaje orientado a objetos como es Java, como parte de su formación inicial en

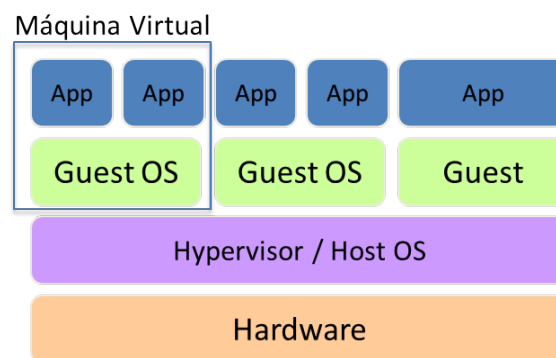


Figura 1: Estructura de una plataforma virtualizada.

la construcción de sistemas informáticos. En el curso 2012/2013 hay 417 alumnos matriculados. Como entorno para realización de prácticas se utiliza CentOS y el entorno de programación BlueJ 3.0.8 sobre JDK (Java Development Kit) 7, para la construcción de algoritmos.

3.2. Virtualización

La virtualización es el mecanismo que permite que un SO anfitrión (*Host OS*) ejecute por encima un SO invitado (*Guest OS*), dentro de una máquina virtual, por medio de la ayuda de un hipervisor (o Virtual Machine Monitor). El hipervisor se encarga de controlar el acceso al hardware subyacente y, por lo tanto, posibilita que múltiples máquinas virtuales se ejecuten sobre un SO anfitrión. Esto permite desacoplar las aplicaciones del hardware subyacente y posibilita la construcción de aplicaciones virtualizadas (*Virtual Appliances*) que agrupan la instalación de un SO junto con todo un conjunto de aplicaciones, de manera que se satisfagan los requisitos software de un determinado entorno de trabajo.

La Figura 1 resume las características de una plataforma virtualizada. Se observa en el nivel más bajo el hardware y por encima se ejecuta el SO anfitrión junto con un hipervisor sobre el que se ejecutan diferentes máquinas virtuales (que agrupan el SO y las aplicaciones). Existen en la actualidad numerosos hipervisores, como es el caso de VMware⁵, KVM [4], Xen [7] o VirtualBox [9].

Para la construcción de EVAP se ha optado por el hipervisor VirtualBox, un producto de virtualización de arquitecturas tanto x86 como AMD64/Intel64, soportado por Oracle y que está disponible como código abierto bajo licencia GPL. VirtualBox es multiplataforma y se ejecuta en Windows, Linux, OS X y Solaris (en el SO anfitrión). Soporta una gran variedad de

⁵<http://www.vmware.com>

SO invitados, como las diferentes variantes de Windows, GNU/Linux, Solaris, OpenSolaris, OpenBSD, etc. Además, dispone de una cómoda interfaz gráfica de usuario que facilita la interacción con el hipervisor.

3.3. Características básicas de EVAP

Antes de describir las características de EVAP es importante destacar que su uso no es imprescindible para realizar las actividades propuestas en la asignatura. Se ofrece como material complementario para el alumno que desea una forma fácil y cómoda de acceder a un entorno de trabajo similar al de los laboratorios. De hecho, el principal software disponible en EVAP puede ser instalado directamente de forma manual por el alumno en su equipo, en cualquiera de los principales sistemas operativos (Windows, GNU/Linux y OS X).

La máquina virtual, creada con la herramienta VirtualBox, consta de las siguientes características hardware:

- Disco duro de 80 GBytes de los cuales cerca de 5 GBytes están en uso para albergar la instalación tanto del SO como de las aplicaciones instaladas.
- 1024 MBytes de RAM. El alumno puede incrementar la memoria asignada a EVAP dependiendo de la memoria RAM disponible en su equipo. Esto permite aumentar las prestaciones de la misma para aquellos equipos más potentes. Se recomienda un mínimo de 2 GBytes de RAM en el equipo físico para ejecutar EVAP de forma fluida.

Estas son las principales características software:

- Sistema operativo GNU/Linux CentOS versión 6.3. Esta es la distribución usada en los laboratorios de prácticas.
- Java SE Development Kit (JDK) 7 Update 13. Contiene el entorno de Java necesario para la compilación de los programas de los alumnos.
- BlueJ versión 3.0.8. Este entorno de desarrollo se caracteriza por su potencia y simplicidad y es especialmente útil para el aprendizaje de la programación.
- Gnuplot. Permite visualizar datos y se utiliza de forma ocasional en prácticas para representar comparativas de eficiencia de algoritmos.

La máquina virtual se configura con acceso a un usuario sin privilegios para evitar que el alumno pueda de forma accidental eliminar la configuración de la misma, aunque se le comunica al usuario la contraseña del usuario *root*. No obstante, una de las ventajas de usar una máquina virtual es que si se produce algún cambio accidental en la configuración de la misma, no hay más que descargarla nuevamente y restaurar así la funcionalidad original.

Para poder utilizar EVAP, el estudiante únicamente tiene que instalar VirtualBox, disponible para los principales sistemas operativos y cuya instalación es muy sencilla para cualquier persona con conocimientos a nivel usuario de informática. El fichero que contiene la máquina virtual (*EVAP.vdi*) ocupa 4.6 GBytes pero puede comprimirse a 1.8 GBytes. Este fichero debe ser descargado por el alumno a su equipo antes de poder poner en marcha EVAP con VirtualBox. Aunque pueda parecer un tamaño de fichero excesivo, la realidad es que con la popularización de la banda ancha, es posible descargar ese fichero en poco más de 23 minutos con una conexión ADSL de 8 Mbit/s y poco más de 1h30' con una conexión de 2 Mbit/s.

Se realizó una guía de instalación y uso⁶ y la distribución de la misma se hizo a través del servidor web departamental.

4. Características avanzadas de EVAP

Además de las características básicas, se apostó por la integración en EVAP de otras dos tecnologías utilizadas en el ámbito de las asignaturas. Por un lado, un corrector de programas para que el alumno pueda realizar de forma autónoma ejercicios y éste software le ofrezca retroalimentación sobre sus soluciones. Por otro lado, vídeo-ejercicios didácticos donde el profesor plantea y resuelve un problema, grabado en vídeo. A continuación se describen ambas herramientas.

4.1. Corrector automático de programas autocontenido

Tanto en las asignaturas de programación de primer curso IIP y PRG como en Estructuras de Datos y Algoritmos (EDA) se utiliza un software, denominado Corrector Automático de Programas (CAP) [8]. Esta herramienta consta de un Applet Java que presenta al alumno problemas típicamente de implementación. El alumno puede compilar su código desde la herramienta y ésta le orienta sobre la corrección de la solución. Para ello, es necesario realizar un programa de test para cada ejercicio, encargado de verificar la solución del mismo y de mostrar al estudiante orientación sobre cómo solucionar los fallos existentes en su implementación. La herramienta utiliza una base de datos MySQL, alojada en un servidor departamental, para almacenar ejercicios así como las soluciones parciales que van realizando los alumnos.

La Figura 2 muestra una captura de pantalla de la herramienta. Se accede al CAP a través de Internet mediante un navegador desde un equipo que tenga insta-

⁶Disponible en <http://goo.gl/QQIid>

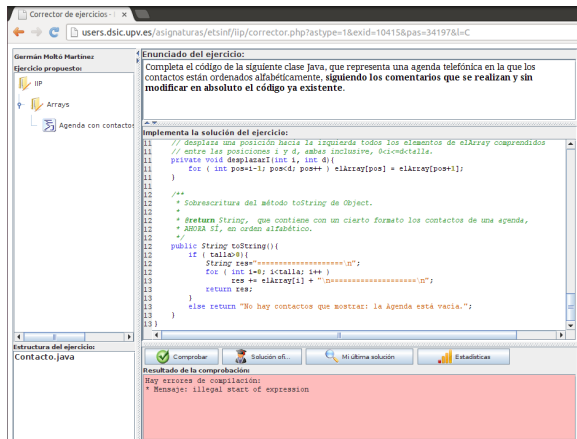


Figura 2: Captura de pantalla del Corrector Automático de Programas (CAP).

lado Java JDK (ya que la compilación del código del usuario se realiza en su propio equipo). Actualmente hay numerosos ejercicios desarrollados para el CAP, tanto para IIP como para PRG. Por ello, es interesante integrar esta herramienta dentro de EVAP con el objetivo de no precisar acceso a Internet para su utilización y garantizar su funcionamiento con independencia de la configuración del equipo del estudiante.

La integración del CAP en EVAP se realizó mediante la instalación de un servidor MySQL en la máquina virtual así como la migración de los datos de la base de datos departamental a dicha instancia de base de datos en EVAP. Fue necesario anonimizar la información recuperada, puesto que el sistema registra las soluciones parciales y las vincula a estudiantes matriculados en las asignaturas. Se crearon los diferentes ficheros HTML que contienen la configuración del Applet para cada ejercicio y éstos fueron integrados en EVAP. La implementación inicial precisaba de código PHP, que generaba los HTML específicos para cada ejercicio. Sin embargo se trató de evitar el despliegue de un servidor web (por ejemplo Apache), con soporte para PHP en EVAP, para no incrementar los requisitos de espacio en disco ni la cantidad de memoria necesaria para el mismo.

Esto permite acceder al CAP desde EVAP con un navegador web que se les deja a los alumnos preconfigurado con soporte para Java. Esto permite realizar los ejercicios propuestos en el CAP sin necesidad de conexión a Internet, y sin interferir en la base de datos departamental. Por tanto, cualquier estudiante, independientemente de que esté matriculado en las asignaturas involucradas, puede utilizar esta herramienta para enfrentarse a ejercicios de programación en Java.

4.2. Vídeo-ejercicios didácticos

En las asignaturas anteriormente mencionadas también existe un amplio trabajo previo de producción de material audiovisual que denominamos vídeo-lecciones. Las vídeo-lecciones son material audiovisual educativo donde el profesor explica un concepto/ejercicio con una duración del orden de 10 minutos, para tratar de mantener la atención del estudiante. En esta línea, el autor ha trabajado en la producción tanto de Polimedias como de vídeo-ejercicios.

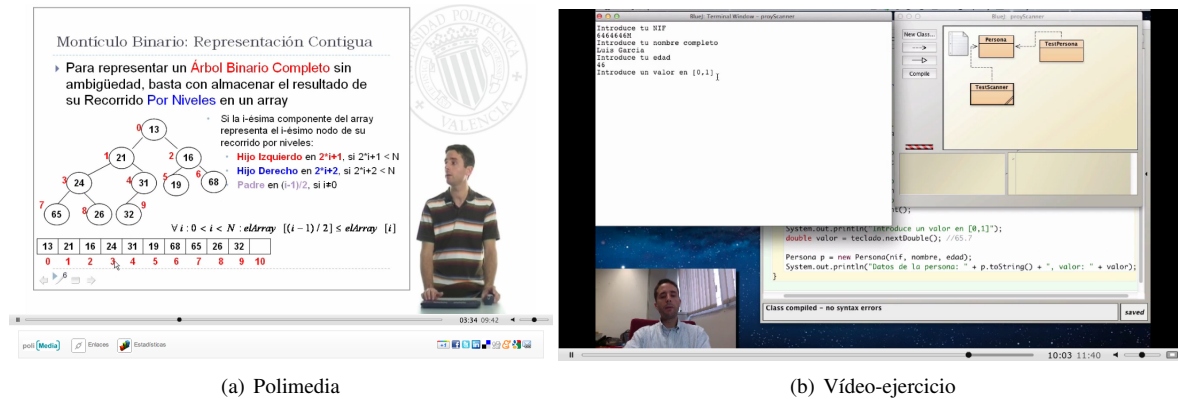
Los Polimedias (ver Figura 3.a) son vídeo-lecciones en las que aparece tanto el profesor como un soporte multimedia (diapositivas, etc.), con el objetivo de guiar la explicación. Éstos se graban en unas instalaciones específicas habilitadas por la universidad y deben grabarse de forma ininterrumpida. Como alternativa flexible, los vídeo-ejercicios (ver Figura 3.b) son vídeo-lecciones en las que el profesor graba el proceso de resolución de un ejercicio en un lienzo digital. Para ello se utilizan técnicas de captura de pantalla (*screen-cast*), una tableta digital o un entorno de programación y un equipo portátil con webcam y micrófono. Algunos ejemplos de vídeo-ejercicios son trazas de ejecución de algoritmos, generalmente recursivos, o diseños de clases Java, para iniciarse en la programación orientada a objetos.

Las vídeo-lecciones tienen muy buena acogida entre los alumnos. Concretamente, de los más de 27 vídeo-ejercicios creados por el autor, que acumulan más de 12600 visitas, más del 95 % de los alumnos (90 encuestas) valoran como útiles para su aprendizaje este tipo de material [5]. Estos vídeo-ejercicios se distribuyen por *streaming* desde PoliTube, la plataforma corporativa de la UPV de difusión de material audiovisual educativo. Sin embargo, los alumnos a menudo preguntan por la posibilidad de descargar los mismos, para tenerlos archivados y poder consultarlos sin necesidad de acceso a Internet.

Con este fin, la idea fue introducir los vídeo-ejercicios en EVAP para que pudieran ser accedidos en local, sin necesidad de Internet. Para ello, se desarrollaron diferentes webs que recopilan los vídeo-ejercicios y los clasifican por tema para cada una de las asignaturas. Esto incluye todo el material relacionado con cada vídeo (clases Java involucradas, transparencias adicionales, etc.). Es posible encontrar un ejemplo accesible online de una de estas webs para la asignatura IIP⁷.

A partir de esta web, resultó muy sencillo construir una versión *offline* de dicha web e incorporarla a EVAP. El único inconveniente es el tamaño de los vídeos, ya que cada uno de ellos ocupa entre 70-100 MBytes y dependiendo del número de vídeos puede aumentar bastante el tamaño de EVAP. Por ejemplo, en IIP actual-

⁷<http://www.grycap.upv.es/gmolto/iip.php>



(a) Polimedia

(b) Vídeo-ejercicio

Figura 3: Ejemplos de vídeo-lecciones. Los vídeo-ejercicios se graban en el equipo del profesor mientras que los Polimedia se producen en un estudio de grabación.

mente hay 10 vídeo-ejercicios disponibles, lo que supone incrementar el tamaño de EVAP en algo menos de 1 GByte.

Para facilitar el proceso de integración, se creó un *shell-script* que descarga de forma recursiva la web que indexa los vídeo-ejercicios, procesa los enlaces a los vídeo-ejercicios y los descarga desde PoliTube. Posteriormente, reemplaza las URLs en las páginas HTML descargadas para que referencien los archivos de vídeos locales, permitiendo al alumno tener una versión idéntica a la web online de vídeo-ejercicios, pero accesible sin necesidad de acceso a Internet.

Como detalle técnico adicional, se utiliza Google Chrome para poder visualizar los vídeos (desde la página web local, referenciados mediante la etiqueta *video* de HTML 5), ya que los vídeos están en formato MPEG-4 y Firefox no soporta dicho formato por cuestiones de licencia.

4.3. Apariencia de EVAP

En la Figura 4 se muestra la apariencia de EVAP, donde el estudiante está visualizando un vídeo-ejercicio sobre cómo diseñar clases en Java (en segundo plano, con el navegador Google Chrome). En primer plano, a la izquierda está el Corrector Automático de Programas (CAP), sobre el que el alumno está resolviendo un ejercicio de implementación de una clase sencilla en Java. El CAP guía al alumno y le orienta sobre los errores de su código para facilitar la labor de resolución del ejercicio propuesto. A la derecha, el alumno tiene el entorno de programación BlueJ para poder desarrollar la clase Cuadrado (por si no quiere utilizar directamente el CAP para ello). Esta máquina virtual, basada en Linux, se está ejecutando por encima del SO del estudiante, que en este caso es Windows.

Obviamente, la integración del CAP y los vídeo-ejercicios en EVAP lleva asociado un incremento en

el tamaño de la imagen de máquina virtual. Esta versión precisa 7 GBytes de espacio en disco y puede ser comprimido a un fichero de 3.8 GBytes.

4.4. Objetos de aprendizaje auto-evaluables

Al unir las tecnologías anteriormente mencionadas (plataforma de desarrollo, vídeo-lecciones y Corrector Automático de Programas) se da la conjunción propicia para la creación de Objetos de Aprendizaje Auto-Evaluables (OA2E), una herramienta para el aprendizaje autónomo de destrezas de programación. A continuación se describe un posible escenario de uso:

El estudiante revisa la guía didáctica [2] del tema actual, que incluye la distribución y temporización de actividades por sesiones, incluidas las actividades no presenciales. El alumno revisa la actividad no presencial correspondiente para dicho día, que le indica que debe visualizar una vídeo-lección sobre la clase Scanner de Java y luego resolver un ejercicio propuesto para verificar la comprensión de los conceptos explicados.

El alumno arranca EVAP, abre un navegador y se conecta a la página (que puede ser offline, almacenada en el propio EVAP, u online), para visualizar la vídeo-lección. En ella, el profesor de forma virtual le explica los conceptos. Posteriormente, el alumno accede al ejercicio propuesto en el CAP para resolver el problema propuesto: escribir un pequeño programa que usa la clase Scanner de Java para leer datos introducidos por teclado. El CAP le orienta en el proceso de resolución, destacando los fallos que comete de manera que el alumno iterativamente va resolviendo los problemas hasta que el CAP indica que su código funciona correctamente. En la siguiente sesión, el alumno acude a clase con esos conceptos ya trabajados, lo que le permite seguir con mayor facilidad la explicación del profesor. De esta manera, el alumno ha podido aprender

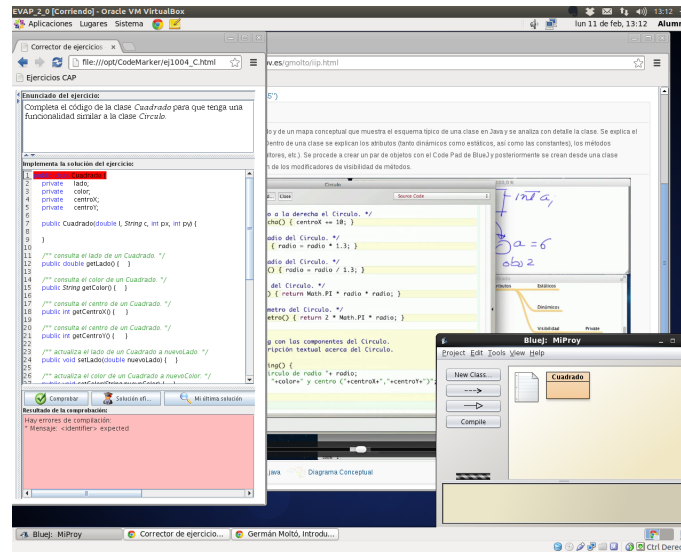


Figura 4: Captura de pantalla de EVAP.

y verificar el aprendizaje de los conceptos de manera desatendida, sin la intervención del profesor.

Por supuesto, este procedimiento también puede realizarse online, sin necesidad de EVAP, pero la integración de las herramientas en una misma entidad ofrece cohesión y posibilidad de uso sin necesidad de Internet, además de garantizar un entorno de trabajo con las herramientas necesarias para la asignatura. Esto no solo es una característica que puede ser deseada por nuestros alumnos, sino que puede resultar de interés para aprendices de países donde el acceso a la banda ancha no sea tan extendido. La posibilidad de realizar una única descarga que contiene un entorno completo para el auto-aprendizaje puede resultar útil.

5. Resultados y discusión

El curso 2012/2013 ha sido el primero en el que se ha incorporado EVAP en la asignatura IIP (primer cuatrimestre de primer curso). De los 417 alumnos matriculados, 109 alumnos la descargaron previa contestación a un pequeño cuestionario online para caracterizar a los usuarios y sus equipos. Los resultados de este cuestionario son los que se muestran en la introducción del artículo. Tras finalizar la asignatura, se les propuso rellenar una sencilla encuesta de satisfacción online que fue contestada por 21 estudiantes (hubieron usuarios de EVAP que no contestaron la encuesta, al no ser obligatoria). Los resultados se refieren a la versión de EVAP que no contiene integrado el CAP ni los video-ejercicios, sino tan solo el entorno de trabajo similar al de laboratorio (características básicas). A continuación se resumen los resultados obtenidos:

El Cuadro 1 resume los resultados de la encuesta. La

primera pregunta trata de averiguar si el proceso de instalación y utilización de EVAP fue de utilidad para los alumnos. La mayor parte de los alumnos (90 %) contestaron afirmativamente (o neutral). No obstante, para la siguiente versión se recopilarán las sugerencias de mejora de los alumnos sobre la guía de la instalación con la idea de incrementar todavía más el porcentaje de satisfacción. La segunda pregunta está relacionada con los posibles problemas técnicos derivados de usar EVAP. En efecto, a menudo la complejidad técnica de un nuevo servicio puede poner en riesgo las ventajas pedagógicas que éste pueda aportar. Por ello, es muy importante que la componente tecnológica ofrezca la mínima fricción para que el estudiante únicamente perciba las ventajas ofrecidas por la nueva herramienta. Aquí, aproximadamente la mitad de los alumnos comentaron que habían tenido algún problema técnico que dificultó (aunque no impidió) utilizar el servicio. Algunos alumnos destacaron la lentitud de la máquina virtual, si bien este problema es derivado en gran medida por las prestaciones del equipo del alumno. En el documento de instalación se especifican los requisitos mínimos recomendados para ejecutar EVAP pero será conveniente advertir a los alumnos de que no lo ejecuten con un número elevado de aplicaciones abiertas en su propio SO, de cara a dejar suficiente RAM libre para que EVAP puede ejecutarse sin problemas.

Con respecto al grado de uso de EVAP a lo largo de la asignatura, un 33 % de los alumnos la usó de forma regular en la asignatura, para llevar a cabo las actividades propuestas. Finalmente, se preguntó a los alumnos sobre el grado de utilidad de EVAP para facilitar su aprendizaje en la asignatura. Éste es quizá uno de los aspectos más importantes, el hecho de que los alumnos perciban como útil ofrecer ese servicio. En este caso,

Pregunta	Escala	Resultados
La explicación del proceso de instalación y uso ha sido clara	[1,5], 1=TED, 5=TDA	10 % en [1,2], 23 % en [3], 67 % en [4,5]
He tenido problemas técnicos que han dificultado usar este servicio	[1,5], 1=TED, 5=TDA	43 % en [1,2], 27 % en [3], 30 % en [4,5]
¿Cual ha sido tu grado de uso de este servicio a lo largo de la asignatura?	Uso esporádico / regular	52 % esporádico, 33 % regular, 15 % poco usada
Indica el grado de utilidad de este servicio para facilitar tu aprendizaje en la asignatura	[1,5], 1=Inútil, 5=Muy útil	20 % en [1,2], 18 % en [3], 62 % en [4,5]

Cuadro 1: Resultados de la encuesta de evaluación de EVAP. La mayoría de las preguntas se expresan en escala [1,5]. TED equivale a Totalmente en Desacuerdo mientras que TDA representa Totalmente de Acuerdo.

el 80 % expresó que consideraba como útil o muy útil la existencia de este servicio en el contexto de la asignatura.

En general, los resultados obtenidos y la retroalimentación recibida por parte de los alumnos invita a continuar con esta línea de trabajo. Con la experiencia adquirida en esta actividad piloto se procederá a mejorar el servicio para las siguientes asignaturas involucradas (principalmente PRG y EDA). Esto permitirá refinar tanto EVAP como la documentación asociada con el objetivo de que un mayor número de alumnos se anime a utilizarlo y sea útil para su aprendizaje.

6. Conclusiones

Este artículo ha presentado los resultados de una experiencia piloto en la creación de entorno virtual de aprendizaje de la programación. Este involucra la creación de una máquina virtual preconfigurada con el entorno que utilizan habitualmente los alumnos en los laboratorios de prácticas, con el objetivo de que los alumnos (de primer curso de primero), puedan tener acceso a un entorno de trabajo propio de forma rápida y sencilla, sin necesidad de lidiar con detalles de instalación y configuración de software. Ese trabajo se complementa con la inclusión de vídeo-ejercicios didácticos y la integración de la herramienta de Corrección Automática de Programas en EVAP, de manera que el alumno disponga de un entorno de aprendizaje autónomo para poder trabajar de forma offline, sin necesidad de acceso a Internet.

EVAP se encuentra a disposición de la comunidad⁸ para ser utilizada por otros estudiantes interesados.

Agradecimientos

El autor desea agradecer a l'Escola Tècnica Superior d'Enginyeria Informàtica (ETSINF) el soporte recibiendo para la difusión de este trabajo.

do para la difusión de este trabajo.

Referencias

- [1] David J Barnes and Michael Kölling, *Objects first with Java: A practical introduction using BlueJ*, Pearson Prentice Hall, 2006.
- [2] Jon Ander Gómez, Marisa Llorens, Carlos Herro, and Germán Moltó, *La Guía Didáctica como Eje Central en la Implantación de Metodologías Activas*, II Congreso Internacional de Docencia Universitaria (II CIDU 2011), 2011.
- [3] Pedro Pablo Gómez Martín, *Máquinas Virtuales en las clases de Informática*, XII Jornadas de enseñanza universitaria de la informática (JENU), 2006.
- [4] A Kivity, Y Kamay, and D Laor, *KVM: the Linux virtual machine monitor*, Proceedings of the Linux Symposium (2007), 225–230.
- [5] Germán Moltó, *Producción y Uso de Vídeo-Ejercicios Didácticos en Asignaturas de Programación*, XVIII Jornadas de Enseñanza Universitaria de la Informática (JENU 2012), 2012, pp. 255–262.
- [6] Abelardo Pardo, Iria Estévez, Damaris Fuentes, Pablo Basanta, and Pedro J. Muñoz, *Apuntes de Arquitectura de sistemas*, 2012.
- [7] I. Pratt, K. Fraser, S. Hand, C. Limpach, A. Warfield, D. Magenheimer, J. Nakajima, and A. Mallick, *Xen 3.0 and the art of virtualization*, Linux symposium, Citeseer, 2005, pp. 65–77.
- [8] Óscar Sapena, Mabel Galiano, Marisa Llorens, and Nati Prieto, *CAP: Corrector Automático de Programas para la evaluación continua y el aprendizaje autónomo*, V Jornada de Innovación Docente (ETSINF), 2012.
- [9] Jon Watson, *VirtualBox: bits and bytes masquerading as machines*, Linux Journal **2008** (2008), no. 166, 1.

⁸<http://www.grycap.upv.es/gmolto/evap.php>