



SERA: Sistema para la Evaluación y Retroalimentación Automática de Prácticas

Isaac Agudo, Rubén Ríos, Ana Nieto

Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga, Spain
Email: {isaac, ruben, nieto}@lcc.uma.es

Resumen—En este artículo presentamos una sistema modular y altamente configurable que permite no sólo la generación y evaluación automática de prácticas de laboratorio sino también proporcionar una retroalimentación instantánea al estudiante para orientar el proceso de aprendizaje y fomentar su autonomía. Este sistema ha sido integrado dentro de la plataforma Moodle en varias asignaturas del Área de Ingeniería Telemática en la Universidad de Málaga.

Palabras Clave—Reutilización, automatización, prácticas, feedback instantáneo, LTI, Moodle, autonomía estudiante

Tipo de contribución: *Formación e innovación educativa en desarrollo*

I. INTRODUCCIÓN

Entre los diferentes objetivos promovidos por el Espacio Europeo de Educación Superior (EEES) se encuentra impulsar un cambio en las metodologías docentes con el fin de mejorar el proceso de aprendizaje del estudiante. Este cambio de metodologías supone, en la mayoría de casos, una mayor dinamización de la docencia, haciendo al estudiante más partícipe de su propio aprendizaje. En definitiva, el objetivo es orientar la enseñanza a la acción, esto es, a que los estudiantes aprendan haciendo en lugar de ser meros consumidores de información.

Este objetivo se ha materializado en diferentes tipos de metodologías como el aprendizaje basado en proyectos/problemas o las clases invertidas [1], reduciendo el número de horas dedicadas a clases magistrales en favor de un mayor contenido práctico. Esto es especialmente relevante en el contexto particular de las ingenierías, donde los autores de este artículo tienen la mayor parte de su docencia, pero también en otras ramas de conocimiento debido a la implantación de las TIC en todos los ámbitos.

Este cambio de perspectiva, que sin duda es beneficiosa para el estudiante, supone un gran esfuerzo adicional para el docente, que no sólo debe preparar los contenidos teóricos necesarios para la asignatura sino que además se ve inmerso en la constante creación y corrección de prácticas con el fin de que los estudiantes reciban

retroalimentación. Del mismo modo, el docente debe guiar a aquellos estudiantes que encuentran alguna dificultad para finalizar la práctica, ya sea en el aula o fuera de ella. En estos casos, si la retroalimentación del profesor no es inmediata, el estudiante suele perder el interés.

Lamentablemente, esta forma de proceder es sólo abordable cuando el número de estudiantes es relativamente reducido ya que, en otro caso, el docente es incapaz de hacer frente a las necesidades del alumnado sin que esto suponga un exceso de carga de trabajo. Ante esta situación, los autores de este trabajo sondeamos diferentes alternativas que pudieran ser integradas fácilmente en la plataforma de enseñanza virtual de la Universidad de Málaga, actualmente basada en Moodle [2]. Entre ellas se consideró utilizar herramientas como Siete (Sistema de Evaluación Inteligente mediante Tests) [3] y VPL (Virtual Programming lab for Moodle) [4] pero no se adaptaban completamente a nuestras necesidades. Siete por estar principalmente orientado a pruebas adaptativas en función de las respuestas de los estudiantes, algo que queda un fuera de nuestras aspiraciones, y si bien es altamente personalizable, el esfuerzo para la creación de nuevas actividades no era abarcable. VPL al estar más enfocado a actividades de programación y, a pesar de ser también altamente personalizable, su interfaz de usuario lo hace poco amigable para otro tipo de actividades.

Por ello, comenzamos realizando algunas experiencias ad-hoc pero (i) por un lado no era sencillo integrar nuestros desarrollos con el Campus Virtual, al que los estudiantes están acostumbrados y (ii) por otro lado, las prácticas planteadas eran difícilmente reutilizables ya que estaban diseñadas para cumplir un objetivo muy concreto.

A partir de estas primeras experiencias nos planteamos la necesidad de desarrollar un sistema que permitiera el diseño de prácticas de laboratorio reutilizables, auto-evaluables y capaces de ofrecer al estudiante asesoramiento en el caso de que el resultado no fuera el esperado. Además, entre nuestros objetivos irrenunciables se encontraba

que se pudiera integrar con la plataforma Moodle para unificar todos los contenidos y resultados de la asignatura en un único lugar. El resultado de este trabajo es SERA.

A continuación se presenta el contexto en el que se desarrolla nuestra propuesta, respaldada por dos proyectos de innovación educativa financiados por la Universidad de Málaga. Seguidamente, en la sección III presentamos detalles sobre el diseño e implementación de la solución. La sección IV describe el lenguaje creado para la especificación de actividades además de mostrar un ejemplo. La sección V presenta nuestras conclusiones y trabajo futuro.

II. CONTEXTO

El sistema SERA comienza a gestarse en un Proyecto de Innovación Educativa (PIE) [5] centrado en la evaluación basada en hitos, donde se observó la necesidad de automatizar la evaluación de competencias prácticas del alumnado.

La experiencia se ha llevado a cabo en asignaturas de varias titulaciones donde los autores de este artículo imparten asignaturas relacionadas con la Seguridad en Redes. Las asignaturas en cuestión son las siguientes:

- *Seguridad en Redes* del Grado en Ingeniería Telemática (Obligatoria)
- *Seguridad Informática e Informática Forense* del Grado en Criminología (Optativa)
- *Seguridad en Redes y Transacciones Online* del Máster en Dirección y Marketing Digital (Obligatoria)

Aunque estas asignaturas están dirigidas a grupos muy heterogéneos, tanto en el número de estudiantes como en sus conocimientos previos sobre redes e informática, todas están orientadas a iniciar al estudiante en el ámbito la Seguridad de la Información. Por ello, a pesar de que cada asignatura tiene una serie de contenidos específicos a la titulación, comparten una serie de contenidos básicos sobre amenazas, servicios de seguridad, criptografía, protocolos y herramientas de comunicación segura. Entre las prácticas más recurrentes en estas asignaturas cabe destacar:

- Implementación de algoritmos criptográficos
- Gestión de certificados digitales
- Cifrado y firma digital de documentos
- Correo electrónico seguro (S/MIME y PGP)
- Servidor Web seguro (TLS)
- Configuración de Cortafuegos
- Redes Privadas Virtuales (IPSEC)

A pesar de que la orientación y el nivel de profundidad alcanzado en cada una de las prácticas depende del perfil de cada titulación, es interesante observar que las prácticas más complejas se pueden descomponer en actividades más sencillas. Estas prácticas sencillas suelen ser comunes a todas las asignaturas, lo que facilita la reutilización mediante la concatenación de actividades.

En lo que a medios tecnológicos se refiere, para la realización de las prácticas nos encontramos con la dificultad de que es necesario utilizar multitud de entornos y herramientas, lo cual dificulta enormemente el seguimiento y evaluación automática de las mismas. Más aún

si consideramos que entre nuestros objetivos está tener un sistema completamente integrado en Moodle.

III. EL SISTEMA SERA

En esta sección presentamos nuestra solución de manera detallada. En primer lugar se ofrece una visión general del funcionamiento del sistema y a continuación se describe la arquitectura e implementación del mismo.

A. Visión Global

SERA es un sistema para el diseño de prácticas de laboratorio que permite la evaluación de competencias relacionadas con la seguridad en redes, la programación y el diseño de protocolos de comunicación.

El flujo de trabajo con nuestro sistema comienza con la especificación de una actividad por parte del docente. Para ello se crea una plantilla con parámetros que permiten instanciar tareas personalizadas para cada estudiante. El valor de estos parámetros es generado por el sistema SERA utilizando unos módulos específicos para cada tipo de actividad. Los parámetros generados servirán, además, a nuestro sistema para comprobar si los envíos realizados por los estudiantes son correctos.

B. Arquitectura

El sistema SERA consta de tres componentes principales (ver Figura 1):

- *Generador* encargado de la creación de las actividades en función de los parámetros proporcionados por el docente
- *Validador* encargado de comprobar si la actividades son correctas en función de elementos enviados por el estudiantes a través de un formulario
- *Monitor* encargado de validar actividades pero mediante la monitorización de acciones realizadas por el estudiante en el sistema

Estos tres componentes dependen de un conjunto de módulos que proporcionan métodos específicos para cada actividad encargados de la generación de las actividades y su evaluación. Los módulos deben ser programados por el docente y podrán ser utilizados en diferentes actividades. Esto permite la modularidad y extensibilidad de SERA.

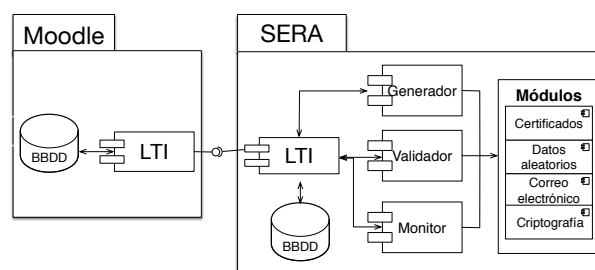


Figura 1: Componentes principales de SERA

Tal como se aprecia en la figura, el sistema SERA está conectado con Moodle a través del estándar LTI (*Learning Tool Interoperability*)¹, ampliamente utilizado

¹<http://www.imsglobal.org/activity/learning-tools-interoperability>

por la comunidad educativa y que ya se ha utilizado con éxito para integrar actividades de programación en Moodle [6].

El estándar LTI permite conectar de forma fácil y segura aplicaciones externas con los sistemas de gestión de aprendizaje (LMS - *Learning Management Systems*) más comunes hoy en día. Mediante LTI se establece una conexión segura entre ambas partes que permite (a) recibir en la aplicación externa, en nuestro caso el sistema SERA, la sesión del estudiante y (b) enviar a la plataforma de aprendizaje, en nuestro caso Moodle, las calificaciones correspondientes.

C. Implementación

Para la implementación de SERA se ha usado *Node.js* y en particular el marco de trabajo para aplicaciones web *express*, que nos permite crear rutas de acceso al sistema. En nuestro caso definimos:

- *lti*. Se encarga de recibir las peticiones de Moodle, verificar los datos del estudiante y llamar al componente Generador para crear una sesión local para el estudiante dentro del sistema SERA. En esencia, transforma las plantillas definidas por el docente (ver sección IV) en actividades.
- *solve*. Se encarga de validar el formulario de entrada del estudiante. Comprobará que el estudiante ha enviado todos los datos solicitados en la actividad y verificará si se cumplen las condiciones especificadas en la actividad.
- *monitor*. Se encarga de proporcionar una conexión entre los monitores de actividad de cada módulo y la interfaz web mediante AJAX. De esta forma, cuando el Monitor detecte que se ha completado la actividad, la interfaz web se refresca y muestra la información necesaria al estudiante.

La funcionalidad del sistema recae en última instancia en los módulos específicos para cada tipo de actividad.

Todos los módulos deben implementar una función de generación de parámetros de forma obligatoria, así como una serie de funciones de comprobación. En el código 1 podemos ver la estructura de esta función para el caso particular del módulo x509.

Código 1 : Función de generación del módulo x509

```

1 async function generateParam(template) {
2   let param = {};
3   if (template.hasOwnProperty('type'))
4     {
5     let subject = template.input.subject;
6     switch(template.type.toLowerCase()) {
7     case 'csr': // Creamos un CSR
8       param = {pem: await pki.createPKCS10(subject),
9                 subject: subject};
9       break;
10    case 'p12': // Creamos un Cert
11      param = {pem: await pki.createPKC12(subject),
12                subject: subject};
12      break;
13      ...
14    }
15  }
16  return param;
17 }
```

Esta función es la que invocará el componente Generador cuando tenga que instanciar alguna actividad que utilice dicho módulo. En este caso, el módulo permite generar un CSR y un certificado (y clave privada) en formato PKCS12 para ser utilizados en la actividad que se quiera definir.

En cuanto a las funciones de comprobación, estas permiten comprobar propiedades de los envíos de los estudiantes, así como comprar los parámetros generados anteriormente frente a los envíos de los estudiantes.

IV. ESPECIFICACIÓN DE ACTIVIDADES

La especificación de las actividades se descompone en 4 bloques:

- **Descripción.** Además de un título, cada actividad debe incluir una descripción que se personalizará para cada estudiante en función de los parámetros generados por SERA.
- **Envíos.** Las actividades suelen solicitar al estudiante el envío de algún tipo de contenido usando un formulario. De no ser así, se debe especificar la función que monitorizará las actividades realizadas por el estudiante.
- **Parámetros.** De cara a personalizar la actividad, el docente puede especificar un conjunto de parámetros que se instanciarán usando los módulos específicos. Hay tres tipos de parámetros:
 - *Literales.* Cadenas de texto personalizadas en función de la sesión del estudiante y otros parámetros generados en la misma actividad.
 - *Referencias.* Referencias a envíos o parámetros de otras actividades ya superadas por el estudiante.
 - *Generados.* Parámetros creados usando módulos específicos.
- **Comprobaciones.** Por último, la actividad debe incluir una serie de comprobaciones que relacionen los envíos del estudiante con los parámetros generados previamente.

Esta forma de especificar actividades permite (a) que cada estudiante reciba actividades personalizadas con distintos valores y (b) establecer dependencias para construir actividades complejas a partir de otras más simples.

Por limitaciones de espacio, en lugar de describir el lenguaje de especificación de actividades en detalle, se muestra un ejemplo de uso.

A. Ejemplo de uso

En el código 2 se puede ver un ejemplo de especificación de actividad. En ella se aprecia como la descripción (línea 3) utiliza código HTML con unas etiquetas especiales que permiten incluir los parámetros de la actividad (líneas 6-12). Entre los parámetros, se incluyen tanto parámetros literales (línea 9) como parámetros generados (línea 11). También se puede ver los campos que debe enviar el estudiante para resolver la actividad (líneas 4-5).

Por último se especifican las comprobaciones (líneas 13-19) de la actividad, que devuelven en caso de fallo

```

Código 2 : Ejemplo de especificación de actividad
1 {id: 20,
2  title: 'Crear CSR para S/MIME',
3  desc: `Crea un Certificate Signing Request (CSR)
      para pedir un certificado de correo electrónico
      poniendo en el CN tu nombre, es decir: <span
      class= "param"><%= cn %></span>. No olvides
      especificar también en el Subject Alternative
      Name tu correo electrónico <span class= "param"
      ><%= email %></span>. Es importante que la
      longitud de las claves que utilices sea de <
      span class= "param"> <%= length %></span> bits.
      `
4  submissions: [
5    {name: 'csr', label: 'CSR', type:'file'}],
6  params: [
7    {name: 'email', value: '<%= eMail %>'},
8    {name: 'san', value: 'email:<%= eMail %>'},
9    {name: 'length', value: '2048'},
10   {name: 'type', value: 'smime'},
11   {name: 'random', module: 'data', type:'dec',
      input:{low:0,high:99}},
12   {name: 'cn', value: '<%= fullName %> <%= random
      %>'},
13  checks: [
14   {module: 'x509', check:'checkSecureHash', input:{
      csr: 'csr'}},
15   {module: 'x509', check:'isCsr', input:{csr: 'csr'
      }},
16   {module: 'x509', check:'checkType', input:{csr: '
      csr', type:'type'}},
17   {module: 'x509', check:'checkCN', input:{csr: '
      csr', CN:'cn'}},
18   {module: 'x509', check:'checkSan', input:{csr: '
      csr', san:'san'}},
19   {module: 'x509', check:'checkKeyLength', input:{
      csr: 'csr', length:'length'}},]]

```

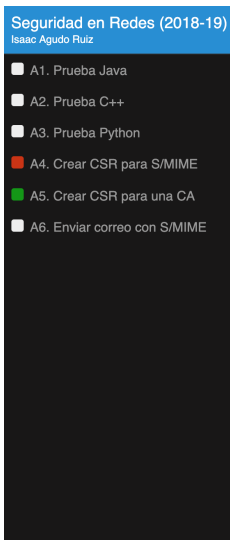


Figura 2: Interfaz Web de SERA

una retroalimentación que puede orientar al estudiante en la consecución de su objetivo. El módulo utilizado para las comprobaciones es el encargado de proporcionar la retroalimentación específica.

En la figura 2 se muestra la vista del estudiante a partir de la plantilla descrita anteriormente. En el menú de la izquierda se muestran las actividades disponibles y se indica, con un código de colores, aquellas que aún no se han empezado (blanco), aquellas se han finalizado con éxito (verde) y aquellas que se han intentado de forma fallida (rojo). En la descripción de la actividad, se puede ver como cada parámetro se ha instanciado de forma particular (en azul) de acuerdo a la especificación y como aparece un formulario de entrega con un único campo, CSR, tal como se indicaba en la plantilla (línea 5).

El estudiante tiene libertad para elegir qué actividad realizar. Las actividades con algún parámetro de tipo referencia, como la actividad A6, requieren que se realicen previamente las actividades que la bloquean.

V. CONCLUSIONES Y TRABAJO FUTURO

En este artículo hemos presentado SERA, un sistema integrado con la plataforma Moodle mediante LTI, que ofrece al estudiante retroalimentación instantánea sobre las actividades que está desarrollando, guiándole en todo momento hacia la solución de manera autónoma.

El sistema está en estado de pruebas en asignaturas de diferentes titulaciones y esperamos seguir ampliando la experiencia a otras asignaturas del área, así como el número de módulos disponibles para permitir que cualquier

docente pueda definir sus propias actividades en SERA sin necesidad de programar un módulo específico.

Estamos en proceso de obtener impresiones de los estudiantes que lo han utilizado y comparar los resultados de las prácticas obtenidos por estudiantes de años anteriores. Aunque aún no disponemos de esos datos, las primeras impresiones por parte del alumnado son muy positivas porque les permite tener una visibilidad clara de su progreso y trabajar de manera más autónoma. Nuestro objetivo es que SERA permita reducir la carga docente asociada a la evaluación y seguimiento continuo.

AGRADECIMIENTOS

Este trabajo ha estado parcialmente financiado por el proyecto de innovación educativa PIE17-137 de la Universidad de Málaga.

REFERENCIAS

[1] R. Clark, A. Kaw, Y. Lou, A. Scott, and M. Besterfield-Sacre, "Evaluating blended and flipped instruction in numerical methods at multiple engineering schools," *International Journal for the Scholarship of Teaching and Learning*, vol. 12, no. 1, 2018.

[2] S. Kumar, A. K. Gankotiya, and K. Dutta, "A comparative study of moodle with other e-learning systems," in *2011 3rd International Conference on Electronics Computer Technology*, vol. 5, April 2011, pp. 414–418.

[3] R. Conejo, E. Guzmán, E. Millán, M. Trella, J. L. Pérez-De-La-Cruz, and A. Ríos, "Siette: A web-based tool for adaptive testing," *Int. J. Artif. Intell. Ed.*, vol. 14, no. 1, pp. 29–61, Jan. 2004. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1434852.1434855>

[4] J. C. Rodríguez-del Pino, E. Rubio Royo, and Z. Hernández Figueroa, "A virtual programming lab for moodle with automatic assessment and anti-plagiarism features," in *Proceedings of The 2012 International Conference on e-Learning, e-Business, Enterprise Information Systems, & e-Government*, 2012.

[5] D. Nunez, F. Moyano, A. Nieto, J. J. Ortega, I. Agudo-Ruiz, and J. López, "A milestone-driven approach for lab assignments evaluation in information security," in *International Conference on e-Learning 2014*, 2014.

[6] A. J. Sierra, Á. Martín-Rodríguez, T. Ariza, J. Muñoz-Calle, and F. J. Fernández-Jiménez, "LTI for interoperating e-assessment tools with LMS," in *Methodologies and Intelligent Systems for Technology Enhanced Learning*. Springer International Publishing, 2016, pp. 173–181.