

# Diseño de un conector para Moodle para analizar el proceso de aprendizaje de tecnologías de acceso a servicios web

Eduard de Torres, Daniel Amo, Oriol Pando, Joan Navarro y Xavi Canaleta  
Departamento de Ingeniería – La Salle Campus BCN  
Universitat Ramon Llull  
Barcelona

{eduard.detorres, damo, oriol.pando, jnavarro, xavic}@salle.url.edu

## Resumen

Un gran número de aplicaciones informáticas modernas utilizan servicios o Interfaces de Programación de Aplicaciones (APIs) hospedadas en servidores remotos. Por esta razón, es habitual basar algunos de los ejercicios prácticos de las asignaturas de las titulaciones de Informática en estas tecnologías para que, así, los estudiantes se familiaricen con este tipo de entornos. La necesidad de enfrentarse y practicar en estos escenarios es incuestionable. Este tipo de entornos presentan un problema y una oportunidad; por un lado, la logística asociada a diseñar, configurar, y mantener el servidor/servicios remotos es muy tediosa. Por otro lado, el hecho de que todos los estudiantes tengan que utilizar un determinado servicio, podría permitir al equipo docente monitorizar de forma automática el uso y progreso de los estudiantes. Por esta razón, en este póster se presenta un conector para Moodle el cual permite (1) crear servicios remotos hospedados en el propio Moodle y (2) calcular y mostrar estadísticas de uso de estos servicios—ya estén hospedados en el propio Moodle o bien en otros servidores remotos. Esto permitiría incorporar estas estadísticas en el sistema de calificaciones. Se describe la política de uso, arquitectura del conector y posibilidades educativas. A modo de ejemplo, se muestra el desarrollo de la herramienta para ser utilizada en la asignatura de Programación Web del tercer curso del Grado en Informática.

## Abstract

A large number of modern computer applications use services or Application Programming Interfaces (APIs) hosted on remote servers. For this reason, it is common to use these technologies on practical exercises on courses from the Computer Science degree. That way students become familiar with this type of programming environment. The need to face and practice in these scenarios is unquestionable. This type of environment presents both a problem and an oppor-

tunity. On one hand, the logistics associated with designing, configuring, and maintaining the remote service(s) is very tedious. On the other hand, the fact that all students have to use a certain service(s) could allow the teaching team to automatically monitor the use to know how students progress. For this reason, this poster introduces a Moodle connector which (1) allows the creation of remote services hosted in Moodle itself, (2) calculates and displays usage statistics for these services—whether they are hosted in Moodle itself or on other remote servers. This would allow these statistics to be used in the rating system. The poster also describes usage policy, connector architecture, and assessment possibilities. As an example, the poster shows the development of the tool to be used in the Web Programming course of the third year of the Degree in Computer Engineering.

## Palabras clave

Entorno virtual de aprendizaje, Interfaz de Programación de Aplicaciones, Análítica del aprendizaje, Computación en la nube, Servicios web.

## 1. Introducción

El auge de la computación en la nube ha hecho que el Software como Servicio (SaaS) se haya convertido en una alternativa muy común para la distribución de herramientas digitales. Estas herramientas tratan datos ubicuos, los cuales se deben acceder de forma remota—lo que requiere de profesionales cualificados en esta disciplina. El contexto en el que se emplaza este trabajo va íntimamente ligado al proceso de enseñanza-aprendizaje del acceso y tratamiento de datos ubicuos en entornos de computación en la nube.

Para satisfacer esta demanda de mercado, en la universidad se enseña a los estudiantes un conjunto de tecnologías para acceder y tratar información que reside en entornos distribuidos (p.ej., computación en la

nube) [6]. En esta enseñanza se requiere procesos de aprendizaje que impliquen comprender y desarrollar soluciones informáticas mediante el uso de tecnologías de acceso a datos ubicuos en la web.

Generalmente, estos procesos son efectivos cuando están mediados por pedagogías y metodologías de aprendizaje situado (PMAS), como el aprendizaje basado en retos. Para el caso concreto de la computación en la nube, se suelen utilizar servicios web de terceros disponibles en la industria, lo que da realismo al proceso de aprendizaje. No obstante, en este contexto situado y externalizado, los docentes pierden la capacidad de hacer un seguimiento, evaluación y retroacción adecuados, a tiempo real y de información actualizada. Tal y como se expuso en [1], resulta muy interesante poder obtener los datos generados por las interacciones del estudiante mientras este está desarrollando una práctica que usa tecnología de acceso a servicios web.

En este póster, se propone ampliar el trabajo desarrollado en [1] y analizar los datos que se generan en prácticas sobre infraestructuras distribuidas (SaaS). Concretamente, se propone una capa intermedia en el uso de tecnologías de acceso a servicios web, que permita al docente comprender cómo los estudiantes están usándolas, y que asegure un nivel adecuado de la privacidad de estos según lo establecido en el Reglamento General de Protección de Datos (RGPD). En esta capa se incorporan procesos de analítica educativa para la captura y análisis de las interacciones de los estudiantes en el uso de estas tecnologías. Con este trabajo se pretende analizar la necesidad de la propuesta con una entrevista a docentes interesados a partir de la cual diseñar un conector para Moodle.

## 2. Metodología

Este trabajo se ejecuta a partir de una definición metodológica multimétodo: (1) Se realiza una revisión de la literatura de los congresos JENUI. Se pretende detectar inquietudes similares y soluciones ya presentadas, (2) Se realizan entrevistas semiestructuradas a docentes interesados. Se comprueba el alcance de las inquietudes de los autores de este trabajo, así como se obtiene retroalimentación para el desarrollo de la solución. (3) Se definen políticas, la arquitectura e interfaz de la solución mediante ingeniería del software (ver Figura 1), (4) Se plantea proceso de usabilidad y experiencia de usuario para evaluación de la interfaz. La ejecución del proceso está en curso.

### 2.1. Revisión de artículos JENUI

Se ha centrado el análisis en ponencias y recursos docentes presentados en ediciones anteriores de las JENUI, para detectar inquietudes similares y soluciones

The image shows a Moodle configuration page for adding a new API. The page title is "Se está añadiendo una nueva API en Sección". It is divided into several sections:

- parámetros generales:** Includes a text input for "Nombre de la API", a text input for "URL de la API" (with a sample URL: "https://study.salleurl.edu/modapi/view.php?id=478454"), and a file upload area for "Ficheros adicionales" with a maximum size of 800MB. A dashed box indicates where to drag and drop files.
- API externa (terceros) - Third party API:** Includes a "URL real" input field and a "Habilitar" checkbox.
- Mapping de parámetros de la petición GET:** A table with two columns: "Nombre del parámetro" and "Nombre real del parámetro". It lists "param1" and "param2" with corresponding input fields and a "ANADIR PARAMETRO" button.
- Mapping de parámetros de la petición POST:** A similar table and button for POST requests.
- API interna (propia) - First party API:** A section with three buttons: "GUARDAR CAMBIOS Y VUELVE AL CURSO", "GUARDAR CAMBIOS Y VISUALIZA", and "CANCELAR API".

At the bottom, a note states: "Este formulario contiene campos obligatorios marcados con \*".

Figura 1: Conector propuesto

ya presentadas. En concreto, se han revisado los últimos cuatro volúmenes de actas publicadas por AENUI sobre los congresos JENUI del 2016 al 2019. Durante este proceso de análisis, se ha percibido una evolución sobre los intereses de los ponentes. En los primeros volúmenes (JENUI 2016 - 2018), se observan ponencias relacionadas con Entornos Virtuales de Aprendizaje (EVAs) que buscan (1) proporcionar recursos docentes de aprendizaje o autoaprendizaje en la nube [2, 8, 14], (2) proporcionar recursos de autocorrección para el uso del estudiante y/o del profesorado [2, 3, 4, 5, 13] y (3) proporcionar recursos para la virtualización de plataformas o escenarios [6, 11]. La mayoría de estos recursos tratan la interacción con el estudiante desde el punto de vista puramente académico, y no profundizan en la extracción de analíticas sobre los comportamientos de dichos estudiantes. Por otro lado, en 2016 se encuentra una ponencia que trata la inquietud de obtener analíticas sobre los estudiantes, en este caso, para combatir el abandono de estos [7]. Llegados al 2019, aún se aprecian ponencias y recursos docentes relacionados con los escenarios anteriores [10, 15], pero se empieza a apreciar un cambio en el enfoque de estas. Finalmente, hay recursos docentes que se centran en conseguir analíticas de aprendizaje de los estudiantes para evidenciar el desarrollo de sus proyectos [1, 9] o para detectar anomalías en las entregas [12].

## 2.2. Políticas y arquitectura ingenieril

Los contextos de enseñanza y aprendizaje en los que se ven envueltos los procesos de analítica del aprendizaje son sensibles y complejos. La exposición de datos de los estudiantes en el uso de herramientas que recolectan y analizan datos, metadatos y datos personales de los estudiantes implica su automática pérdida de control si no se actúa con cautela y responsabilidad. Esta pérdida de control y gestión de datos es a todos los niveles, de estudiante, de profesor y de institución educativa. El uso de Interfaces de Programación de Aplicaciones (APIs) de terceros viene acompañado de una responsabilidad para el docente, y la institución educativa, de velar por una adecuada privacidad de sus estudiantes. La solución que se presenta en este trabajo respeta la identidad y datos del estudiante en el uso de APIs de terceros, como base ética y moral. Del mismo modo, los objetivos del desarrollo del conector parten de la privacidad del estudiante desde el diseño y por defecto pretende (a) asegurar la no exposición de datos de los estudiantes, (b) asegurar el control y la gestión de los datos generados por los estudiantes, docentes e institución educativa, (c) gestionar la API desde el mismo EVA, (d) crear una API propia de manera fácil y rápida, (e) enlazar, asegurar y parametrizar una API de terceros, (f) facilitar el seguimiento, la tutoría y la evaluación de cada uno de los estudiantes en el transcurso de la asignatura, (g) incrementar la seguridad en el uso de la API (sólo la pueden utilizar aquellos alumnos inscritos al curso) y (h) incluir la API dentro de la dirección web del EVA de la institución educativa.

En consideración a las premisas anteriores, se propone APIProxy<sup>1</sup> ya que su función es la de ser un intermediario entre el estudiante y una API de terceros. A modo de *man in the middle*, pero en connotación positiva, este conector facilita que (1) los estudiantes puedan usar la API sin ver expuesta su identidad ni datos personales, (2) los docentes puedan crear enlaces a API de terceros e incluso una API propia con total independencia del soporte tecnológico de la institución, (3) se puedan recolectar datos asociados a las interacciones de los estudiantes, individualmente, para ser usados en procesos de analítica del aprendizaje.

El conector se presenta a los docentes dentro de Moodle como un recurso Actividad, gestionable mediante un formulario de fácil uso y de interfaz gráfica mimetizada con el diseño de Moodle para facilitar su adopción. La arquitectura del conector es compleja debido a la necesidad de autenticación: para evitar usos fraudulentos y para poder hacer el seguimiento del estudiante. En la Figura 2 se muestra un diagrama de la arquitectura desgranada en 4 flujos de datos. En cada flujo de datos interviene una información concreta pa-

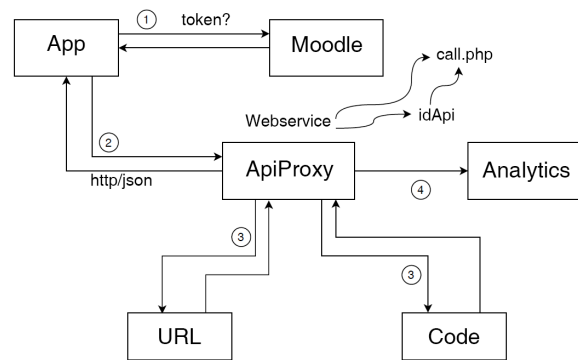


Figura 2: Diagrama de la arquitectura del conector

ra hacer posible el funcionamiento del conector: Flujo 1: La App que desarrolle el estudiante y use la APIProxy, requiere de autenticación con Moodle, para generar el token de usuario que usará en las peticiones a APIProxy. Flujo 2: Peticiones a la APIProxy en métodos GET y POST. Flujo 3: Petición a la API de terceros o al código fuente que el docente quiere que se ejecute. Flujo 4: Recolección de las peticiones a las API del flujo de datos 3 relacionadas con cada estudiante.

## 3. Resultados preliminares

De las entrevistas a roles educativos se extrae que APIProxy es adecuada, válida para satisfacer las necesidades e inquietudes y que cumple con las políticas expuestas en la Sección 2. El número de profesores entrevistados es 5, en una implicación de 7 asignaturas de La Salle - Universitat Ramon Llull, donde todas las respuestas coinciden en la necesidad de capturar las interacciones de los estudiantes con la API para conocer comportamientos de uso. Para evaluar la solución propuesta, se ha utilizado en la asignatura de Programación Web 1 del Grado en Ingeniería Informática de La Salle - Universitat Ramon Llull, en la cual están matriculados 83 estudiantes. Los resultados obtenidos están reflejados en el Cuadro 1. El docente puede introducir la configuración del recurso e indicar, tanto si la API es de terceros como propia, los valores que deben enviarse para que la petición sea correcta.

## 4. Conclusión

Se demuestra en el trabajo que es posible solucionar la necesidad de comprender cómo los estudiantes interactúan con API tanto de terceros como propias. La solución en formato conector es la más adecuada en cuanto a instalación técnica y uso. El conector actúa como agente intermedio, permitiendo proteger la identidad del estudiante. Se añade que todo acceso al

<sup>1</sup> Disponible en [github.com/oriolpando/ApiProxy](https://github.com/oriolpando/ApiProxy)

Tipo de error	Nº
Número de peticiones	8245
Malformed strings	217
Parámetros incorrectos	979
Valores incorrectos	641
Errores internos de la API externa	57

Cuadro 1: Errores totales más frecuentes recogidos en el log de APIProxy

conector queda recolectado en el log de Moodle, generándose un resumen de accesos y errores de uso por estudiante que fomentan posibilidades educativas como soporte al seguimiento o evaluación. Como trabajo futuro se deja pendiente de realizar un proceso de experiencia de usuario y usabilidad con los docentes para evaluar la validez de la interfaz gráfica del conector.

## Referencias

- [1] Dani Amo, Joan Navarro y Xavi Canaleta. «Analizando el proceso de aprendizaje de nuestros alumnos: un caso práctico». En: *Actas de las JENUI*. 2019, págs. 357-360.
- [2] Jesús Aransay, Francisco García, Jónathan Heras, Adrián Inés y Gadea Mata. «Creación de un servidor de integración continua para gestión y corrección de entregas de prácticas». En: *Actas de las JENUI*. 2018, págs. 111-118.
- [3] Pedro Delgado-Pérez, Inmaculada Medina-Bulo y Daniel Pérez-Caro. «Biblioteca CAC++ para la corrección automática de prácticas de programación en C++». En: *Actas de las JENUI*. 2017, págs. 297-304.
- [4] Jose Divasón, Ana Romero y Eduardo Sáenz-De-Cabezón. «Experiencia con una herramienta para automatizar la comprobación de requisitos de los trabajos HTML-CSS de Sistemas Informáticos». En: *Actas de las JENUI*. 2018, págs. 173-180.
- [5] Pedro Pablo Gómez-Martín y Marco Antonio Gómez-Martín. «!Acepta el reto!: juez online para docencia en español». En: *Actas de las JENUI*. 2017, págs. 289-296.
- [6] Jesus González, J Damian Segrelles y Germán Moltó. «Grid as a Service: Herramienta para el despliegue y gestión de un Grid en la nube para actividades educativas». En: *Actas de las JENUI*. 2017, págs. 305-311.
- [7] Carmen Lacave, Ana I Molina, Miguel A Redondo y Manuel Ortega. «Redes bayesianas para identificar perfiles de estudiante. Aplicación al estudio del abandono de las titulaciones de Informática en la Universidad de Castilla-La Mancha». En: *Actas de las JENUI*. 2016, págs. 85-92.
- [8] Erick Orlando Matla Cruz, Miguel Ehécatl Morales Trujillo y David Velázquez Portilla. «QueryCompetition: un sistema web para practicar consultas en SQL». En: *Actas de las JENUI*. 2017, págs. 313-320.
- [9] Germán Moltó, Diana M Naranjo y José Ramón Prieto. «Herramienta web para el seguimiento automatizado de actividades educativas prácticas en la nube». En: *Actas de las JENUI*. 2019, págs. 175-182.
- [10] Guillermo Quintana, Héctor Pérez, Mario Aldea, Julio Medina y Rubén Sebrango. «Extensión de Moodle para la evaluación automática de ejercicios de programación en Java». En: *Actas de las JENUI*. 2019, págs. 151-157.
- [11] Francisco J Ribadas-Pena, Rubén Anido-Bande y Víctor M Darriba-Bilbao. «DSBOX: herramienta docente para el diseño y simulación de entornos de red virtualizados». En: *Actas de las JENUI*. 2016, págs. 335-342.
- [12] Juan Ramón Rico-Juan, Antonio-Javier Gallego, Francisco J Castellanos y Jorge Calvo-Zaragoza. «Sistema para la detección temprana de anomalías en la evaluación usando técnicas de aprendizaje automático». En: *Actas de las JENUI*. 2019, págs. 311-318.
- [13] Miguel Ángel Rubio y Francisco González del Valle. «Uso de una herramienta de corrección automática en un curso de programación: Una experiencia docente». En: *Actas de las JENUI*. 2018, págs. 345-350.
- [14] Juan Ruiz de Miras, Daniel Expósito Porras, David Rocha Cerezo y M<sup>a</sup> Robles Ortega. «Herramienta Web para la propuesta, realización y corrección interactiva de ejercicios de informática gráfica». En: *Actas de las JENUI*. 2016, págs. 303-310.
- [15] Fernando Sáenz Pérez. «DESweb: una herramienta para el aprendizaje de SQL». En: *Actas de las JENUI*. 2019, págs. 95-102.