

Evidentia: gestión de evidencias en trabajos colaborativos usando software libre

David Romero

Dpto. de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
drorganvidez@us.es

David Benavides

Dpto. de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
benavides@us.es

Resumen

El aprendizaje cooperativo consiste en el trabajo conjunto entre estudiantes siendo el docente un mediador del proceso. Existe un reto relacionado con individualizar las notas de una tarea realizada por varios alumnos en estos contextos. En una asignatura se viene implementando un proyecto cooperativo para lo cual, entre otras actividades, se han celebrado ya 9 ediciones de unas jornadas tecnológicas organizadas por los estudiantes de manera colaborativa en donde se presentan ponencias, talleres, *escape rooms* entre otros. Las jornadas, aunque pueden tener una valoración global, tienen que permitir evaluar de forma individual a los estudiantes para así cumplir con la normativa de calificaciones. Desde hace varios cursos, se utiliza un sistema basado en evidencias que permite individualizar la nota según las horas y calidad del trabajo. Sin embargo, este trabajo de custodia, cómputo, calificación y gestión de las evidencias se hacía tedioso y costoso. En las últimas tres ediciones, se ha desarrollado, implementado y desplegado una aplicación web llamada Evidentia, creada por los propios estudiantes de la asignatura que automatiza muchos de los procesos de gestión de evidencias. Evidentia facilita el seguimiento y calificación individual en tareas colaborativas y además permite la personalización para otro tipo de escenarios de enseñanza cooperativa.

Abstract

Cooperative learning consists of joint work among students, with the teacher being a mediator of the process. There is a challenge related to individualizing the grades of a task carried out by several students in these contexts. In one subject, a cooperative project is being implemented for which, among other activities, 9 editions of technological conferences organized collaboratively by students have already been held, where presentations, workshops and *escape rooms*, among others, are presented. The conferences, although they may have a global assessment, must allow students to

be assessed individually in order to comply with the grading regulations. For several years, an evidence-based system has been used, which allows individual marks to be awarded according to the hours and quality of the work. However, this work of custody, computation, grading and management of the evidence became tedious and costly. In the last three editions, a web application called Evidentia has been developed, implemented and deployed, created by the students of the course, which automates many of the evidence management processes. Evidentia facilitates individual monitoring and grading in collaborative tasks and also allows customization for other types of cooperative teaching scenarios.

Palabras clave

Jornadas docentes, software libre, evidencias, innovación docente, proyectos cooperativos

1. Introducción

Desde los años 80 ha existido una revolución en la era de la información [2]. Diversos sucesos han promovido el cambio en la educación; los nuevos enfoques sobre cómo impartir la docencia han facilitado el aprovechamiento de las nuevas tecnologías en la labor de la enseñanza. Esto es, en términos de la impartición de clases y la organización de actividades entre los estudiantes que hace que se complemente su formación. Ha existido un cambio continuo en los paradigmas de aprendizaje, desde las clases magistrales, estático, con poca adaptabilidad al cambio, hasta modernas vías de docencia, incluyendo el uso de recursos docentes multimedia, desde simples *tutoriales* grabados hasta aplicaciones web interactivas con objeto de transmitir ese conocimiento [1].

Dependiendo de la forma que tienen los estudiantes para organizarse de cara a realizar las actividades propuestas por el profesor, nos encontramos con tres tipos principales de aprendizaje [20]. El primero, *el aprendi-*

zaje individualizado, en el que los estudiantes realizan trabajo autónomo sin interactuar con el resto. El segundo, el *aprendizaje competitivo*, donde los alumnos compiten para obtener la mejor calificación, implicando que el éxito de un alumno conlleva el fracaso de otro. Finalmente, el *aprendizaje cooperativo*, donde las actividades se realizan entre todos los alumnos cooperando entre sí, de tal forma que no es posible el éxito sin que el resto de compañeros también lo obtengan.

El aprendizaje cooperativo representa una forma de poner en alza el modelo constructivista de la educación [6]. El aprendizaje colaborativo, por un lado, se centra exclusivamente en crear grupos de alumnos que trabajan para un objetivo en común, bien mediante la realización de tareas, exposición de algún proyecto entre otros. Por otra parte, el aprendizaje cooperativo, más que reunir a un grupo de personas, busca generar sinergias entre ellas, intentando que el conocimiento de unos pocos sea compartido por muchos, donde los estudiantes más aventajados puedan apoyar a los más rezagados, fomentando un clima de participación y motivación apoyado por todos y entre todos [13].

Un problema común que aparece en el trabajo colaborativo intrínseco al aprendizaje cooperativo es la evaluación. Siempre es objeto de debate cómo realizar una evaluación individual en una tarea que viene siendo global de tal forma que sea justa, equilibrada y que evite engaños por parte de estudiantes que se nutren del trabajo de los demás para justificar el suyo propio. Además, es importante cumplir con las normativas de calificaciones y evaluaciones de las universidades que, por lo general, tienen establecida una nota individual para cada estudiante.

Desde hace varios cursos escolares, en una asignatura de 4º del Grado de Ingeniería del Software, se viene implantando un enfoque colaborativo en la enseñanza. Para ello se establece una serie de actividades para conformar la nota global de cada estudiante. Entre estas actividades está la organización de unas jornadas tecnológicas cuya gestión recae principalmente en los estudiantes, siendo el profesor un guía y mediador en el proceso. Con esta actividad se pretende fomentar la cooperación y sinergia de los estudiantes. Esta aproximación tiene la dificultad enunciada antes, es decir, requiere de un seguimiento y calificación individual para un trabajo cooperativo. Las jornadas han sido en general un éxito, y cada año que se organizan van teniendo más calidad, más aceptación y más visibilidad, tanto interna como externa. Sin embargo, cada estudiante debe tener al final una nota individual para poder calcular su nota global y así también cumplir con los reglamentos y normativas existentes.

Para poder calificar el trabajo individual de cada estudiante desde hace unos años se estableció un sistema basado en evidencias de modo que cada estudiante de-

bía elaborar, custodiar y entregar un portfolio [4] de *evidencias*. La tarea de gestión de las evidencias -que por lo general se hacía de manera manual- recaía en el propio comité de organización de las jornadas que empleaba muchas horas en dicha gestión lo que, además, implicaba un trabajo tedioso para los profesores encargados de la evaluación, no siendo pocas las incidencias que había que gestionar además del empleo de unas horas menos productivas para el proceso de aprendizaje colaborativo.

Desde hace tres ediciones, se viene usando una herramienta llamada *Evidentia* que permite automatizar muchas de las tareas, además de servir como punto central para la interacción entre estudiantes y profesor en torno a las evidencias de trabajo en las jornadas. Hasta lo que tenemos constancia, esta herramienta es única en el ámbito de la enseñanza colaborativa y ha permitido gestionar de manera exitosa las evidencias de las últimas ediciones siendo además bien valorado por los usuarios. La herramienta está disponible con una licencia de software libre y puede ser personalizada para otro tipo de aproximaciones colaborativas más allá de la organización de unas jornadas puesto que permite definir los grupos de trabajos, las actividades, los roles, los plazos, las puntuaciones y otros aspectos, lo que convierten a *Evidentia* en una herramienta de propósito general. Además, *Evidentia* tiene la particularidad de haber sido concebida, diseñada, desarrollada y mantenida por los propios estudiantes, teniendo cada año aportaciones que la hacen mejorar en cada edición. Hemos llevado a cabo una evaluación del proyecto que nos muestra un promedio de valoración de utilidad del 70 % como herramienta para personalizar las notas en entornos colaborativos.

Este artículo presenta *Evidentia*, el problema que aborda, cómo lo soluciona y sus características principales. Para ello, en la sección 2, *Contexto del problema*, analizamos la forma original de proceder para la evaluación de estas jornadas y los problemas que presentaba; en la sección 3, *Solución tecnológica*, abarcamos una solución concreta que nos soluciona los problemas descritos; en la sección 4, *Impacto observado*, realizamos unas encuestas de satisfacción a los estudiantes con objeto de medir el grado de aceptación de la solución y posibles mejoras; y en la sección 5, *Conclusiones*, concluimos el trabajo.

2. Contexto del problema

2.1. Planteamiento

En una asignatura de 4º curso del Grado en Ingeniería del Software en una universidad se vienen realizando unas jornadas tecnológicas y educativas desde hace 9 años. Son jornadas organizadas exclusivamente por

el alumnado de esa asignatura de manera colaborativa, donde se presentan ponencias, concursos de programación, talleres, *escape rooms*, ponencias, sorteos, entre otras actividades.

La figura del profesor es meramente representativa, asiste a las reuniones con el rol exclusivo de *asesor* o *mediador*, delegando el peso de todas las decisiones en la presidencia o el órgano directivo que los alumnos hayan estimado más oportuno para ese año. La estructura organizativa en el comité de dirección puede variar de un curso a otro algo que también es decisión de los propios estudiantes.

Las jornadas no solo son interesantes desde un punto de vista divulgativo [5]. Estas tienen un peso en la evaluación de la asignatura establecida en el correspondiente programa y proyectos docentes. El problema que se presenta es cómo evaluar de forma individual el trabajo, en este caso colaborativo de los estudiantes, de forma que sea justo y acorde al nivel de implicación y de horas invertidas. No resulta fácil, pues, establecer una dinámica de trabajo que sea reutilizable cada curso, que implique poca o casi ninguna interacción del docente y que no haya fraude en las horas de trabajo, por citar algunos inconvenientes. Por tanto, es necesaria la creación de un sistema de recogida y procesamiento de las horas de trabajo de los estudiantes. Para ello, la evaluación personal de cada alumno que haya decidido participar en las jornadas viene dada por la implicación de horas de trabajo y calidad del mismo; horas que, con posterioridad, el alumno debe evidenciar de forma documental, bien sea con un texto, un póster, un audio o cualquier medio que considere oportuno.

2.2. Recogida y procesamiento de la información

Al igual que las decisiones respecto a las jornadas recaen en el comité de dirección, la organización de cómo se van a gestionar y salvaguardar todas las evidencias también. Solía existir un *grupo* o *comité* encargado de detallar algún trámite, protocolo o similar que debía seguir el resto de alumnos para que puedan ser evaluados por el profesor. Generalmente, esta tarea recaía en la *secretaría general*.

El comité de dirección, conformado por la presidencia y las coordinaciones de comités, junto con la secretaría general, elegían el sistema de recogida de información, que normalmente solía ser una hoja de cálculo compartida acompañada de otros mecanismos más o menos manuales. La secretaría general, que era la encargada de la custodia y gestión de las evidencias, establecía una hoja a modo de plantilla que el resto de comité debía seguir. Los valores de las columnas eran, a grandes rasgos: *nombre y apellidos, horas de eviden-*

cias, horas de reuniones, horas de asistencia a eventos, bonos complementarios de horas, entre otros. Cada fila representaba un estudiante.

Cada comité era responsable de rellenar la hoja de cálculo según la plantilla que había establecido la secretaría general, aunque era habitual que se agregaran/modificaran/eliminaran campos según la especificidad del comité en cuestión. Para la recogida de evidencias, los alumnos debían mandar un correo electrónico al coordinador del comité asociado a su trabajo, que tenían que rellenar según unas directrices del comité de dirección.

Para la recogida de horas de reuniones, los secretarios de cada comité eran responsables de disponer de una hoja de firmas para cada reunión, donde los asistentes firmaban en el acto. El trabajo del secretario era doble; por un lado, debía trasladar esas firmas de asistencias a la hoja de cálculo oportuna; por otro, debía redactar la convocatoria y el acta de reunión, volcando nuevamente las asistencias a esos documentos.

Para la recogida de horas de asistencias, el comité de Logística creaba unos eventos en la plataforma Eventbrite¹. Los alumnos se registraban en los eventos y les llegaba un código *QR* [19] a la app de su smartphone. Cuando se celebraba el evento, el responsable de logística escaneaba ese código *QR* y la asistencia quedaba contabilizada. Sin embargo, las asistencias debían ser exportadas de Eventbrite e importadas a mano en el excel de asistencia.

Para la recogida de bonos de horas, el comité de dirección establecía un número de horas determinado a computar entre los participantes de las jornadas en concepto de comunicación entre comités, labores de gestión, entre otros. Esas horas se veían reflejadas en la hoja de cálculo.

Una vez concluidas las jornadas, todas las horas de evidencias, reuniones, asistencias y bonos debían ser reunidas en una hoja de cálculo final. Esta labor era competencia de la secretaría general, que debía registrar a mano, alumno por alumno, cada tipo de hora. Esta hoja final era entregada al docente para su posterior evaluación.

2.3. Inconvenientes de esta metodología

Esta forma de proceder presentaba los siguientes problemas:

- *Pérdida de información*: el hecho de mandar las evidencias a través del correo electrónico supone un riesgo de pérdida de información (por ejemplo, correos considerados *spam*), de datos erróneos (existen erratas en el documento adjunto y no se subsanan) y dificultad de que exista una co-

¹<https://www.eventbrite.es/>

municación fluida entre la persona que quiere evidenciar su trabajo y el coordinador de ese comité.

- *Sistema tedioso*: es un procedimiento puramente manual. Con los años, distintas aplicaciones como los documentos *online* han permitido facilitar la tarea de poner en común la información al trabajar de forma colaborativa, pero sigue existiendo el inconveniente de establecer cada curso un protocolo de actuación que difiere con el resto de cursos.
- *Escasa trazabilidad*: un problema que presenta esta metodología es que no permite al docente saber el estado del trabajo de las jornadas hasta el fin de las mismas, imposibilitando estudiar y tomar acciones durante el transcurso.
- *Alta probabilidad de cometer errores*: el hecho de tener a un alumno o alumna por triplicado o cuadruplicado en varias hojas de cálculo induce al error casual que luego suele ser difícil de subsanar o detectar.
- *Información distribuida*: el trabajo en horas de un alumno estaba distribuido en varias hojas de cálculo (o el sistema que se hubiera elegido ese año) desconociendo hasta el final el cómputo global de horas de esa persona.

3. Herramienta para la recogida de evidencias en trabajos colaborativos

3.1. Antecedentes

En el curso 2019/20, la secretaría general, conociendo la problemática decidió, por iniciativa propia, crear un sistema de información web que facilitara la recogida y gestión de la información. Tras estudiar varias opciones, se optó por el uso del framework Laravel [10] por la cercanía al lenguaje PHP [14] de otra asignatura de 2º curso.

Dada la inminente celebración de las jornadas y el hecho de disponer de poco tiempo de desarrollo, se optó por una implementación *ad hoc*, siguiendo una metodología similar a la programación extrema [18] al no contar con unos requisitos funcionales detallados desde el origen [17].

El planteamiento original del sistema de información web era tratar de gestionar las horas de evidencias, horas de reuniones, horas de asistencia y bonos de forma que se evitara la duplicidad de la información. Siguiendo un desarrollo basado en entregas [16] se implementó una aplicación llamada *Evidentia* (nombre procedente de una figura retórica [15]), que fue puesta en producción tan pronto como fue posible. Es importante señalar que el desarrollo y uso de la prime-

ra versión de *Evidentia* se hizo al mismo tiempo que se implantaba la solución.

3.2. Visión general del proceso

En la figura 2 se observa el proceso general por el que pasa la herramienta cada curso y cada una de sus etapas.

Un problema que resuelve *Evidentia* es la reutilización del sistema para varios cursos. La herramienta crea una nueva instancia de la base de datos con objeto de convertir cada curso en artefactos independientes, manteniendo un histórico de trabajo. Esto se observa en el proceso 1.

Una vez que se ha realizado la instancia, se procede al registro de estudiantes. Existe ya una funcionalidad para esto, que lee una hoja de cálculo con columnas ya definidas, típicamente el nombre, apellidos, identificadores únicos, entre otros. Esto se aprecia en el proceso 2. Una vez volcada la hoja de cálculo, el sistema crea una cuenta de usuario con el rol de *Estudiante* por defecto. El administrador del sistema, que por norma general suele ser el profesor, designa a algún estudiante como presidente, aunque pueden ser varios. El estudiante tiene competencias para designar nuevos roles y crear nuevos grupos de trabajo, si así lo requiere el caso de uso. Esto se observa en los procesos 3 y 4 de la figura 2.

Una vez hecho esto, el sistema está listo para recibir el trabajo de los estudiantes. La recogida evidencias, validaciones de las mismas, reuniones y asistencias, marcadas en los procesos 5, 6, 7 y 8 respectivamente, se realizan en paralelo. Una vez terminados estos procesos, se realiza el registro de bonos de horas, como se observa en el proceso 9. Una vez han terminado todos los procesos anteriores y las jornadas se den por finalizadas, se realiza el proceso 10, donde se exporta toda la información útil y que se requiera para la evaluación.

3.3. Recogida de evidencias

En *Evidentia*, los estudiantes son los responsables de redactar sus evidencias y adjuntar los archivos que consideren necesarios para justificar el número de horas empleadas. La gestión de evidencias tiene dos características fundamentales:

- *Estados*. Las evidencias pueden poseer varios estados. *En borrador*: para que sean editadas en cualquier momento; *en revisión*, para que pueda ser estudiada, aprobada o rechazada por el coordinador del comité, *aprobada*, si la evidencia ha sido aprobada, o *rechazada*, junto con un motivo que explica la razón de ese rechazo, con la posibilidad de volver al estado *En borrador*.

Figura 1: Vista principal del usuario

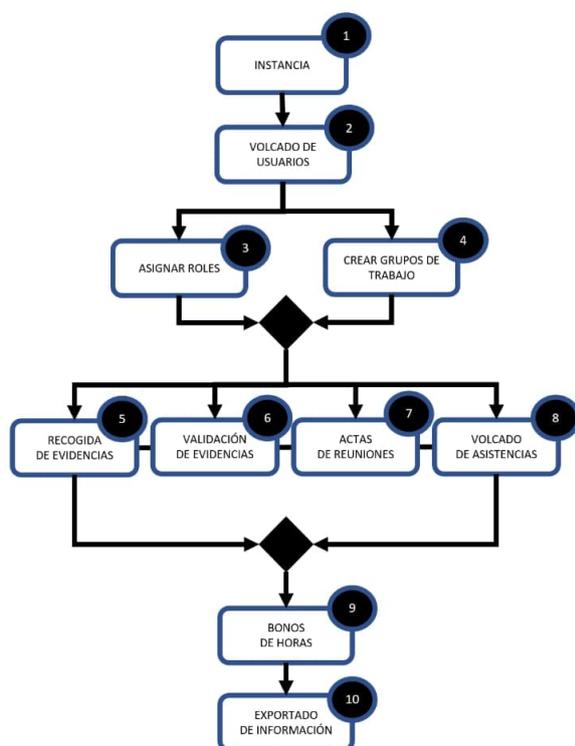


Figura 2: Visión general del proceso

- *Sistema antifraude.* El Espacio Europeo de Educación Superior (EEES) y su sistema de créditos ECTS (European Credit Transfer and Accumulation System) establece una dedicación de 25 horas de trabajo por cada crédito. Es por ello que el recuento de horas es crítico para no sobrepasar ese límite y siga el proyecto docente de la asignatura establecido al comienzo de curso. La información de una evidencia y sus archivos pasan por un proceso de *hash* [11] usando el algoritmo *sha256* [3] Esto genera una cadena de caracteres única, evita que el alumno argumente que su información ha sido modificada y hace que sus horas refleje la dedicación real al trabajo.

3.4. Recogida de horas de reuniones

Los secretarios pueden crear convocatorias y actas de reunión. Las opciones de gestión son:

- *Convocatoria de reunión.* El secretario puede convocar una reunión, avisando por correo electrónico al resto de miembros de su comité. En esta convocatoria es donde se esclarece un orden del día que puede ser reutilizado a la hora de redactar el acta.
- *Hoja de firmas.* Con cada convocatoria de reunión, los posibles asistentes tienen disponible una hoja de firmas virtual. Usando su usuario y contraseña, registran la asistencia sin necesidad de usar papel físico y sin tener que volcar las firmas manualmente, constandingo la fecha y hora.

- *Acta de reunión.* Las horas de asistencia a una reunión se contabilizan aquí. El secretario puede volcar tanto el orden del día de la convocatoria como la hoja de firmas a dicha acta en un proceso automático.
- *Generación automática de documentación.* Las convocatorias y las actas de reunión son generadas automáticamente en formato PDF para ser dispuestas a todo aquel interesado en consultarlas.

3.5. Recogida de asistencia a eventos

El comité de Logística sigue usando el sistema de Eventbrite, detallado en la sección 2.2, para registrar las asistencias. Sin embargo, existen diferencias con el método manual:

- *Consumo de API Rest.* Eventbrite tiene un servicio RESTful [12] que proporciona una API Rest [22] completa para consultar datos. *Evidentia* extrae las asistencias de Eventbrite, realiza una comparativa de nombres y apellidos y registra en su base de datos la asistencia y el evento, junto con el estado, que puede ser *Pendiente de asistir*, *Sin asistencia*, *Confirmado*, entre otros.
- *Exportación de asistencias.* El sistema permite la exportación automática de las asistencias contabilizadas a una hoja de cálculo para ser publicadas al margen del resto de trabajos que se estuvieran llevando a cabo en las jornadas.

3.6. Evaluación

El alumno o alumna con el rol de presidente tiene la capacidad de generar una hoja de cálculo que unifica de forma automática todas las horas empleadas en las jornadas. Esta hoja puede contemplar tanto las evidencias, reuniones, asistencias y bonos como cada concepto por separado, según lo requiera el docente.

Como se muestra en la Figura 1, se ofrece un *dashboard*, accesible tanto por el docente como por los alumnos, que reúne la información más relevante, como el cómputo total de horas, los eventos disponibles a los que asistir, entre otros.

3.7. Desarrollo y mantenimiento

El proyecto *Evidentia* ha sido puesto en producción [7] desde el 2020/21 en adelante. La herramienta es mantenida siguiendo las competencias transversales de la asignatura, como son la gestión del código fuente, la integración y despliegue continuo, el uso de contenedores de Docker, gestión de la configuración, gestión de las incidencias, entre otros. Es por ello que el software *Evidentia* no solo gestiona las evidencias de las

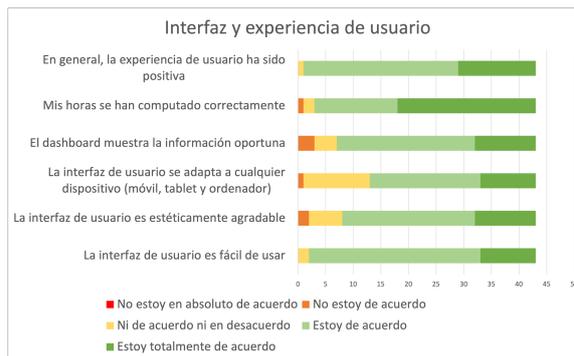


Figura 3: Interfaz y experiencia de usuario

jornadas docentes del curso en cuestión, sino que también ha sido implementado como proyecto evaluable en la asignatura con un tercio de la nota, aproximadamente. *Evidentia* es el único proyecto de todo el Grado que, si bien fue concebido para un curso concreto, son los propios alumnos quienes se encargan de su desarrollo y mantenimiento cada año, dando solución a un problema recurrente. El software puede ser descargado, estudiado y reutilizado desde la siguiente dirección: <https://github.com/drorganvidez/evidentia>

El software *Evidentia* es aplicable en cualquier contexto que requiera de la gestión documental de horas de trabajo, ya sea en jornadas, talleres, conferencias. Usando como base el framework Laravel, sería *relativamente* sencillo incorporar nueva funcionalidad que dé solución a futuros requisitos del software y poder ampliar el uso de *Evidentia* a otros ámbitos de gestión documental.

4. Impacto observado

Al finalizar el curso académico, se les pidió a los alumnos que realizaran una encuesta de satisfacción, con objeto de conocer el grado de éxito de la aplicación, tanto como herramienta de trabajo como proyecto de curso. La encuesta fue realizada por un total de 43 personas, respondiendo a 30 preguntas agrupadas en 5 secciones.

Para medir el grado de aceptación en cada pregunta, se ha usado una escala Likert [23] del 1 al 5, siendo el 1 *no estoy en absoluto de acuerdo* y el 5 *estoy totalmente de acuerdo*.

Las secciones y sus objetivos son:

- *Interfaz de usuario.* Esta sección tiene por objetivo estudiar el impacto de la interfaz y de la experiencia del uso durante el curso académico, como se puede ver en la figura 3.
- *Funcionalidades.* En esta sección hacemos un sondeo de la funcionalidad actual y la percepción

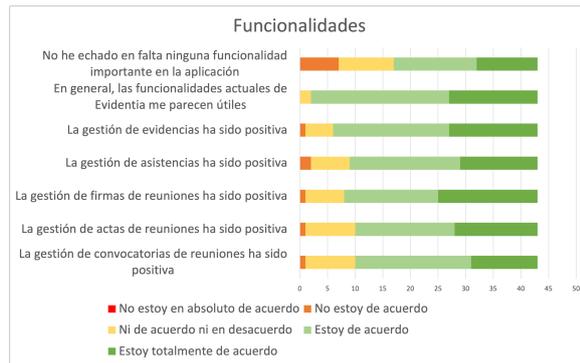


Figura 4: Funcionalidades

de los alumnos de su utilidad, valorando si son eficaces o no, como se observa en la figura 4.

- **Rendimiento y soporte técnico.** Aquí hemos querido estudiar el rendimiento y la disponibilidad de la aplicación, dado que es la única herramienta de gestión que se usa para las jornadas docentes. En la figura 5 se pueden ver esos resultados.
- **Proyecto de curso.** El proyecto, además de servir de herramienta de gestión, también es propuesto como proyecto de curso evaluable, donde grupos de estudiantes tienen que evolucionar el software implementando nueva funcionalidad y documentarlo. En la figura 7 se observa el impacto.
- **Valoración general.** Para concluir, se intenta estudiar, en términos generales, la valoración general del sistema (figura 6), haciendo hincapié en la utilidad de la herramienta como evaluación individual y colaborativa.

En el cuadro 1 destacamos las dos preguntas que poseen más relevancia con este artículo, que son el *tipo de evaluación* y las distintas percepciones, *positiva, negativa, neutra*.

No obstante, en la figura 4 se observa un grado de aceptación menor. Es necesario un análisis en profundidad acerca de qué espera el alumno del uso de la herramienta, qué funcionalidades considera necesarias y están ausentes y cuáles pueden ser mejoradas.

En la figura 5 referente al rendimiento, la cantidad de críticas es debida a la falta de pruebas exhaustivas, (e.g. pruebas funcionales, pruebas de rendimiento, pruebas de integración.). Es necesario, por tanto, hacer hincapié en esta etapa del desarrollo.

Evaluación	P. positiva	P. negativa	P. neutra
Individual	69.76 %	2.36 %	27.91 %
Colaborativo	62.79 %	2.36 %	9.3 %

Cuadro 1: Percepción de la evaluación

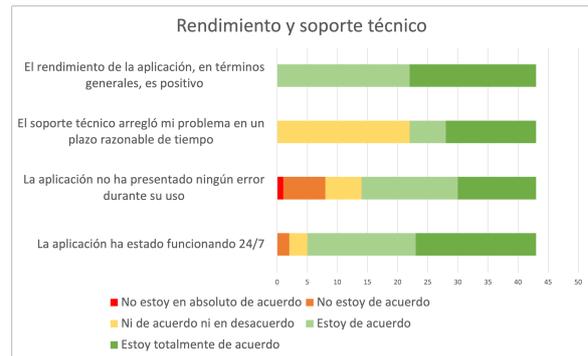


Figura 5: Rendimiento y soporte técnico

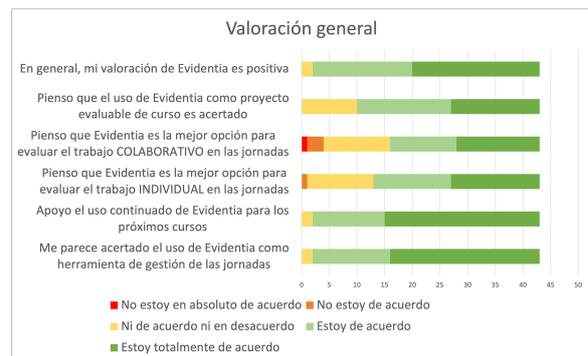


Figura 6: Valoración general

Como evaluación general, en la figura 6 se observa más aceptación de *Evidentia* en la evaluación individual de tareas, con un 69.76 % frente a la evaluación colaborativa, con un 62.79 %, lo que ratifica nuestra idea de que *Evidentia* es una opción válida para este tipo de aprendizaje.

5. Conclusiones

El software *Evidentia* facilita el seguimiento y el cómputo de horas de trabajo y asistencia de los alumnos a las jornadas. Es posible conocer, en cualquier momento, el flujo de trabajo que siguen todas aquellas personas involucradas en la tarea colaborativa de curso. Gracias a esto, la evaluación es más justa y equilibrada, los errores humanos se reducen y permite a los organizadores de las jornadas emplear tiempo y esfuerzo en tareas menos repetitivas y tediosas.

Evidentia es el único proyecto de todas las asignaturas del Grado que es realizado, mantenido y puesto en marcha por y para los alumnos y alumnas, logrando un sistema altamente colaborativo [8]. Creemos que el uso de un software donde el peso del desarrollo recae en el alumnado es una fuente de motivación e interés, amén de estar relacionado con las competencias directas y transversales de la asignatura, e.g. la integración

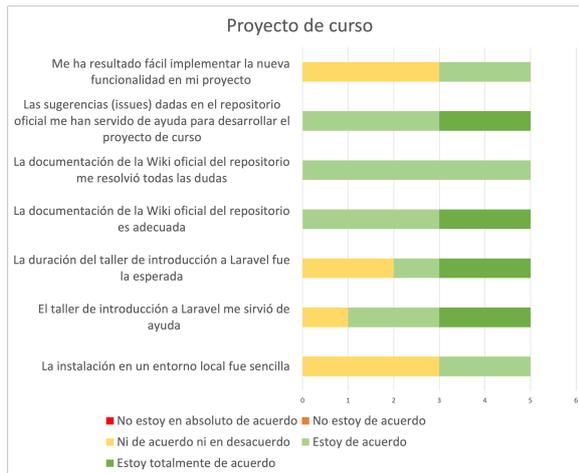


Figura 7: Proyecto de curso

y el despliegue continuo [9], la gestión de código fuente [21] y la automatización de pruebas [24]. Dado que existe, no solo como herramienta para la organización de las jornadas, sino como proyecto de evaluación, esperamos que el software sea evolucionado en los próximos cursos y que su funcionalidad sea extendida y mantenida por los propios alumnos [8].

Referencias

- [1] Jose María Cela-Ranilla, Vanessa Esteve González, Francesc Esteve Mon, Juan González Martínez, y Mercè Gisbert-Cervera. El docente en la sociedad digital: Una propuesta basada en la pedagogía transformativa y en la tecnología avanzada. *Profesorado. Revista de Currículum y Formación de Profesorado*, 21:403–422, 2017.
- [2] Carlos Coello Horna y Alexandra Perez. Revolución digital y trabajo en el S. XXI. 07 2017.
- [3] Dan Michael Cortez, Ariel Sison, y Ruji Medina. Cryptanalysis of the modified SHA256. pp. 179–183, 07 2020.
- [4] John DiMarco. *The Resume, Cover letter, Portfolio Handbook: The comprehensive guide for college students y job seekers*. 01 2022.
- [5] Brayan González, María Calderón, y Daniel Marín. Divulgación científica contemporánea. pp. 19–36, 11 2021.
- [6] Raul Herrada Valverde, Rosario y Baños. Aprendizaje cooperativo a través de las nuevas tecnologías: Una revisión. *@tic revista d'innovació educativa*, p. 16, 06 2018.
- [7] Richard Hicks. *Cloud Deployments*, pp. 175–214. 01 2022.
- [8] Diego Ibarra-Corona, Alexandro Escudero-Nahón, Mauricio Ibarra Corona, y Jose Alejandro Vargas-Diaz. Análisis crítico al concepto de "software educativo": una cartografía conceptual. 05 2021.
- [9] Les Jackson. *The CI/CD Pipeline*, pp. 305–347. 09 2020.
- [10] Krzysztof Kuflewski y Mariusz Dzieńkowski. Symfony y Laravel – a comparative analysis of PHP programming frameworks. *Journal of Computer Sciences Institute*, 21:367–372, 12 2021.
- [11] Marius Mihailescu y Stefania Nita. *Hash Functions*, pp. 83–102. 09 2021.
- [12] Andy Neumann, Nuno Laranjeiro, y Jorge Bernardino. An Analysis of Public REST Web Service APIs. *IEEE Transactions on Services Computing*, 14:957–970, 07 2021.
- [13] Rodolfo Posada Álvarez. Formación superior basada en competencias, interdisciplinariedad y trabajo autónomo del estudiante. *Revista Iberoamericana de Educación*, 35(1):1–33, ene. 2004.
- [14] David Powers. *PHP: A Quick Reference*, pp. 41–80. 01 2022.
- [15] D. Pujante. The discursive construction of reality in the framework of the rhetoric: The constructive rhetoric, 01 2018.
- [16] Simon Robertson. Agile scrum - an overview, 08 2016.
- [17] Pilar Rodríguez, Agustin Yague, Pedro Alarcón, y Juan Garbajosa. Metodologías Ágiles desde la perspectiva de la especificación de requisitos funcionales y no-funcionales. pp. 333–344, 01 2008.
- [18] Anchit Shrivastava, Isha Jaggi, Nandita Katoch, Gupta Deepali, y Sheifali Gupta. A Systematic Review on Extreme Programming. *Journal of Physics: Conference Series*, 1969:012046, 07 2021.
- [19] Alberto Silva y Silas Fantin. A matemática do QR Code. 2:374–399, 12 2021.
- [20] Robert E. Slavin. Cooperative learning. *Review of Educational Research*, 50(2):315–342, 1980.
- [21] Serhat Uzunbayir y Kaan Kurtel. A Review of Source Code Management Tools for Continuous Software Development. *3rd International Conference on Computer Science y Engineering (UBMK)*, pp. 414–419, 2018.
- [22] Balaji Varanasi y Maxim Bartkov. *Introduction to REST*, pp. 1–19. 01 2022.
- [23] Robert Wall Emerson. Likert scales. *Journal of Visual Impairment Blindness*, 111:488–488, 09 2017.
- [24] Francisco Espiga Fernández y Miguel Angel Mainar Angel. Aplicación de técnicas de pruebas automáticas basadas en propiedades a los diferentes niveles de prueba del software. 2015.