



Education in the Knowledge Society
E-ISSN: 2444-8729
fma@usal.es
Universidad de Salamanca
España

Traverso-Ribón, Ignacio; Balderas-Alberico, Antonio; Dodero, Juan Manuel; Ruiz-Rube, Iván; Palomo-Duarte, Manuel

Evaluación sostenible de experiencias de aprendizaje basado en proyectos

Education in the Knowledge Society, vol. 17, núm. 1, 2016, pp. 19-43

Universidad de Salamanca

Salamanca, España

Disponible en: <http://www.redalyc.org/articulo.oa?id=535554761002>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

Evaluación sostenible de experiencias de aprendizaje basado en proyectos

Sustainable assessment of learning experiences based on projects

Ignacio Traverso-Ribón¹, Antonio Balderas-Alberico², Juan Manuel Dodero², Iván Ruiz-Rube², Manuel Palomo-Duarte²

¹ FZI Research Center for Information Technology, Karlsruhe, Baden-Württemberg, Germany. traverso@fzi.de

² Escuela Superior de Ingeniería, Universidad de Cádiz, Cádiz, España. {antonio.balderas, juanma.dodero, ivan.ruiz, manuel.palomo}@uca.es

Resumen

En las experiencias de aprendizaje basadas en proyectos, la monitorización detallada de las actividades de los miembros de cada equipo puede resultar de utilidad para la evaluación de su trabajo. Mediante procedimientos de evaluación, los supervisores pueden evaluar las capacidades de trabajo en equipo con una finalidad formativa. Suelen aplicarse estrategias como la auto-evaluación, la evaluación entre iguales y la co-evaluación para hacer que la evaluación formativa sea sostenible. Seguir una estrategia de evaluación no es siempre sencillo para los miembros de los equipos, pues necesitan un conocimiento razonable del proceso y los criterios de evaluación. Este artículo describe una metodología de evaluación orientada al aprendizaje y un *framework* software de datos abiertos aplicable a la evaluación de proyectos de desarrollo colaborativo. Se han elaborado una rúbrica y una serie de indicadores que proporcionan evidencias sobre las habilidades y, posteriormente, se han aplicado a un curso basado en proyectos de pequeña escala. Los proyectos se gestionaron y ejecutaron con la ayuda de una forja de software libre que incluye una herramienta de gestión de tareas para la planificación y el seguimiento, un repositorio de control de versiones para almacenar el software entregable y una wiki para almacenar los entregables textuales. La experiencia ha proporcionado evidencias a favor del método de evaluación y el *framework* de datos abiertos para hacer la evaluación del trabajo en equipo más sostenible.

Palabras clave

Evaluación del aprendizaje; Aprendizaje basado en proyectos; Forjas de Software; Wikis.

Recepción: 11-11-2015

Revisión: 12-01-2016

Abstract

In a project-based learning experience, the detailed monitoring of the activities in which team members participate can be useful to evaluate their work. Using learning-oriented assessment procedures, supervisors can assess the teamwork abilities with a formative purpose. Evaluation strategies such as self-assessment, peer assessment and co-assessment are often used to make evaluation formative and sustainable. Conducting an assessment strategy is not easy for team members, since they need before to have a reasonable understanding of the evaluation process and criteria. This paper describes a learning-oriented evaluation methodology and an open data framework that can be applied to collaborative project settings. An evaluation rubric and a series of indicators that provide evidences about the developed skills have been elaborated and applied in a small-scale project-based course. Projects were managed and developed with the help of an open source software forge that contains a ticketing tool for planning and tracking of tasks, a version control repository to save the software outcomes, and using a wiki to host text deliverables. The experience provides evidences in favor of using the assessment method and open data framework to make teamwork evaluation more sustainable.

Keywords

Learning assessment; Project-based learning; Software forges; Wikis.

Aceptación: 01-02-2016

Publicación: 01-03-2016

1. Introducción

El análisis del aprendizaje es un área de investigación que tiene por objetivo recolectar, medir, analizar e interpretar datos relativos a los estudiantes en su contexto de aprendizaje (Ferguson, 2012). Uno de los principales desafíos en este área es el desarrollar las habilidades necesarias para proporcionar *feedback* formativo sobre cómo debe ser el aprendizaje de los estudiantes (Ferguson, 2012). Entender el proceso de aprendizaje y los entornos de aprendizajes virtuales, o *Virtual Learning Environment* (VLE), que den soporte a las experiencias de aprendizaje *online* es fundamental para el aprendizaje *online*. Nuestro interés general se centra en investigar cómo las técnicas analíticas de aprendizaje pueden ser implementadas sobre alguna plataforma tecnológica para dar soporte a la evaluación del proceso de aprendizaje.

Los procedimientos de evaluación de aprendizaje se basan en la inspección de un conjunto explícito de productos o resultados de aprendizaje, habitualmente en forma de *portfolios* (Klenowski, Askew & Carnell, 2006), los cuales incluyen un conjunto de elementos que los estudiantes deben desarrollar y entregar en determinados momentos del proceso de aprendizaje. Los miembros de los equipos y los supervisores tienen que examinar un conjunto de aspectos y rellenarlos sobre algún tipo de formulario estructurado o instrumento de evaluación, como las rúbricas o escalas de valoración (Walvoord & Anderson, 2009), con una serie de valores para cada uno de los criterios definidos. El profesor o supervisor puede gestionar el proceso de aprendizaje aplicando procedimientos de evaluación sistematizados e instrumentos (Markham & Larmer, 2003) para evaluar los resultados intermedios y finales, para así poder hacer la evaluación de manera sostenible. Según Boud (Boud, 2000), una evaluación sostenible tiene como objetivo el desarrollo de un conjunto de habilidades que permita evaluar el aprendizaje de los alumnos a gran escala. La evaluación sostenible se basa en la aplicación de estrategias colaborativas, tales como la auto-evaluación y la co-evaluación (Barootchi & Keshavarz, 2002), en las cuales los propios miembros de los equipos llevan a cabo las tareas de evaluación. Esas actividades colaborativas producen una gran cantidad de datos que pueden ser difícilmente tratadas por un único supervisor de proyectos. Esto pone en evidencia un problema importante de escalabilidad, ya que esa evaluación no podría ser abordada si se incrementase el número de integrantes o el total de proyectos a analizar. Las estrategias de evaluación colaborativa pueden aliviar la carga de trabajo de los supervisores/profesores (Falchikov, 2013) mediante la utilización de instrumentos de evaluación que permitan autoevaluar o evaluar entre pares el correcto desempeño de las actividades y de sus resultados (Sadler, 2010). Esto ha demostrado ser de éxito en evaluaciones formativas que incluyen *feedback* y auto-supervisión explícita como parte de las tareas de los estudiantes (Boud, 2007). Una evaluación sostenible ofrece a los estudiantes la confianza

necesaria para progresar en su aprendizaje a lo largo de su vida, sin aumentar la carga de trabajo del personal académico (Davies, 2010).

En el caso de la auto-evaluación y de la evaluación entre pares, suele ser difícil para los miembros de los equipos saber cómo tienen que estimar tales valores y cómo proporcionar las explicaciones necesarias sobre la razón de cada valoración. Desde el punto de vista del supervisor, tales procedimientos de evaluación requieren un esfuerzo considerable si aumenta el número de proyectos, usuarios, grupos o entregables esperados. Eventualmente, la actividad de los proyectos genera una gran cantidad de datos que pueden ser difícilmente evaluados por un único supervisor de proyectos. Esto expone un problema importante de escalabilidad, en el sentido en que la evaluación podría no ser sostenible si el número de usuarios o el número de proyectos se incrementa. No siempre es deseable dejar el procedimiento de evaluación completamente en manos de los miembros de los equipos ya que podrían no estar familiarizados con la evaluación y no tener capacidad de autocritica.

Este trabajo tiene por objetivo intentar ofrecer respuesta a la siguiente cuestión: ¿Cómo podrían los supervisores evaluar de forma sostenible los resultados de experiencias complejas de aprendizaje? La amplia diversidad de procesos de aprendizaje *online* y de entornos de aprendizaje virtuales hace complicado definir metodologías generales de evaluación aplicables de forma extensiva para todos los VLE. Por tanto, el alcance de este estudio se limita a experiencias de aprendizaje basado en proyectos, o *Project-Based Learning* (PBL), que son gestionadas con ayuda de las herramientas facilitadas por las forjas de software. PBL es un tipo de proceso de aprendizaje que se asemeja al contexto de los proyectos de desarrollo de software gestionados mediante las forjas, tales como Assembla o SourceForge. Las forjas de software son herramientas contenedoras, habitualmente basadas en aplicaciones web, que permiten planificar, gestionar, controlar y monitorizar las actividades de un proyecto. Las forjas de software habitualmente incluyen funcionalidades para el control de las tareas a desarrollar y de las incidencias, junto con herramientas wiki para escribir de forma colaborativa los entregables de los proyectos. Asimismo, los sistemas de control de versiones de código se utilizan cuando los resultados del proyecto se basan fundamentalmente en *software* (aunque no siempre es así), permitiendo almacenar y compartir todas las versiones de los entregables de código fuente. El uso de tales herramientas por parte de los usuarios permite descubrir prácticas relativas a las capacidades de trabajo en equipo, entre otras, ya que en las bases de datos internas de las forjas se almacenan las evidencias de tales prácticas.

En este contexto, hemos elaborado algunos indicadores y recomendaciones para ofrecer a los miembros de los equipos un conjunto de pistas e indicios que permitan evaluar ciertas habilidades de forma crítica, mediante el análisis de las actividades de los usuarios registradas en los *logs* de las forjas. De esta manera, se pretende que la evaluación de este tipo de experiencias de aprendizaje

sea más eficiente y sostenible. Proporcionar recomendaciones para la evaluación puede ayudar a mejorar la sostenibilidad de ese proceso (Boud, 2007). Adicionalmente, la utilización de un método que incluya indicadores de evaluación basados en la extracción y agregación de datos de evidencias desde los VLE, podrán ser de ayuda para construir tales recomendaciones, tanto para los profesores como para los estudiantes. Para los profesores, porque puede aliviar la carga de trabajo necesaria para la evaluación, y para los estudiantes, porque puede ayudarles a conocer cómo llevar a cabo los procedimientos de autoevaluación y evaluación entre pares. Se trata de los dos lados de la misma moneda de la sostenibilidad, pero vista desde diferentes perspectivas. Este artículo no se centra en los aspectos pedagógicos de la evaluación del aprendizaje, sino en cómo se puede emplear las aplicaciones web para extraer los datos que permitan llevar a cabo las experiencias de evaluación de aprendizaje asistida por ordenador.

Después de estudiar los trabajos actuales y los fundamentos relacionados con la evaluación en los VLE (véase sección 2), hemos construido un marco de trabajo para la integración de datos (véase sección 4) que permite implementar la propuesta descrita en la sección 3. Nuestra propuesta se basa en el uso de un modelo de datos simple que permita integrar los conjuntos de datos procedentes de diferentes fuentes, con el objetivo de proporcionar recomendaciones de evaluación para el aprendizaje basado en proyectos. Asimismo, se incluye un caso de estudio donde el proceso de aprendizaje PBL utiliza como VLE una serie de aplicaciones existentes en una forja de software libre. Bajo estas restricciones, hemos llevado a cabo el caso de estudio siguiendo un modelo de desarrollo de código abierto, u *Open Source Software* (OSS), el cual se adecua correctamente para experiencias colaborativas de PBL (Zhao & Elbaum, 2003).

2. Fundamentos y trabajos relacionados

Los entornos de aprendizaje virtuales son utilizados extensamente para dar soporte a las experiencias de aprendizaje. Estos entornos abarcan desde los sistemas de aprendizaje o *Learning Management Systems* (LMS), hasta las wikis, blogs o redes sociales. Estos sistemas funcionan como un registro de actividades y recursos disponibles para los participantes en el proceso de aprendizaje, haciendo que la comunicación entre ellos será más sencilla (Zafra, Gibaja, Luque & Ventura, 2011; Munkhchimeg & Baigaltugs, 2013).

Además, esto permite evaluar las competencias de los estudiantes cuando interactúan con las actividades y recursos gestionados bajo los VLE. Cuando el número de estudiantes inscritos en un curso basado en VLE es demasiado alto, los procedimientos de evaluación tradicionales no son nada sostenibles. Un ejemplo de esta situación ocurre cuando se intenta evaluar en el contexto de los

cursos masivos online o *Massive Open Online Courses* (MOOC) (Lugton, 2012; Mor & Koshinen, 2013), los cuales están dirigidos a un número ilimitado de estudiantes. La evaluación del aprendizaje en este tipo de cursos no es sostenible, a menos que se utilicen medios automatizados para ello, o empleando estrategias de evaluación basadas en la autoevaluación o en la evaluación entre pares de estudiantes (Johnson, 2013). En el resto de esta sección se analizan diversas estrategias para la evaluación sostenible del aprendizaje en el contexto de los VLE.

2.1. Autoevaluación y evaluación entre pares

Los *portfolios* o diarios de aprendizaje de los VLE (Klenowski, Askew & Carnell, 2006) se utilizan como instrumentos de evaluación principal. Gracias a ellos, se puede obtener información útil relativa al rendimiento de los estudiantes y de cómo ellos aprenden (Palomares, 2011; Gil *et al.*, 2011). De esta forma, el trabajo de los estudiantes es evaluado mediante una estrategia de autoevaluación o evaluación entre pares, para medir sus progresos y evaluar si han alcanzado o mejorado ciertas habilidades genéricas.

En (Lim, Lai & Ng, 2011) se describe cómo se aplicó esta estrategia a ciertos estudiantes. Los estudiantes que trabajaban conjuntamente en una wiki fueron evaluados por ellos mismos haciendo uso de una rúbrica para comprobar si habían conseguido el nivel de competencias requerido. En (Piedra, Chicaiza, López, Romero & Tovar, 2010; Shih, 2011) se presenta un escenario donde los entornos de aprendizaje fueron integrados con redes sociales. En este trabajo se describe cómo los profesores proporcionaban a sus estudiantes unas rúbricas, que les permitían entender cuáles son las expectativas de los instructores sobre ellos. Posteriormente, los estudiantes daban a sus compañeros la información sobre lo que habían aprendido y lo que todavía necesitaban aprender.

Experiencias parecidas pero utilizando recursos multimedia se recogen en otros trabajos. Por ejemplo, en (Martin-Cuadrado, Lopez-Gonzalez & Garcia-Arce, 2013) se muestra un proyecto donde se realizaban videoconferencias entre estudiantes de una universidad española y una norteamericana. El objetivo era mejorar el aprendizaje y las competencias de comunicación en un segundo lenguaje. En otro trabajo, descrito en (Masip-Álvarez *et al.*, 2013), se describen dos experiencias en la cual los estudiantes eran divididos en grupos y luego exponían sus trabajos ante una videocámara. Luego, estos videos eran subidos al LMS, para que ellos pudiesen observar y corregir sus propios trabajos viendo los fallos que cometían. De esta manera se pretendía evaluar si los estudiantes lograban ciertas competencias.

2.2. e-Evaluación

La e-evaluación se relaciona con aquellas propuestas que describan estrategias novedosas de evaluación usando las nuevas tecnologías. Estas estrategias se basan en el empleo de técnicas de

minería de datos, del rastreo del comportamiento y del análisis de los VLE sobre la interacción de los estudiantes (Redecker, Punie & Ferrari, 2012; Redecker & Johannessen, 2013). Los beneficios de este tipo de estrategias para los procesos de aprendizaje son incluso más sustanciales cuando el método de entrega de los ejercicios por parte de los alumnos es realizado de forma electrónica y de forma automática se lleven a cabo los procedimientos de evaluación (Amelung, Krieger & Rosner, 2011).

Sin embargo, es poco probable que la e-evaluación sustituya por completo a otras formas de evaluación, quedando todavía lejos el día en que el ordenador pueda calificar automáticamente las preguntas de tipo ensayo. De todas formas, resulta interesante investigar en recomendaciones aplicables en las prácticas de e-evaluación y las consecuencias de su aplicación, así como en ofrecer ayuda a la hora de elegir el software correcto, teniendo en cuenta aspectos importantes como la interoperabilidad, escalabilidad, niveles de rendimiento, capacidades de actualización, soporte y mantenimiento, seguridad, accesibilidad y coste (Wood, 2008).

En la literatura pueden encontrarse varios trabajos relacionados con la e-evaluación. Por ejemplo, en (Achcaoucaou, 2014) se presenta la herramienta *Tricuspid*, la cual permite a los estudiantes identificar sus habilidades y carencias y desarrollar estrategias personales de mejora. Además, ofrece a los profesores información adicional sobre cómo esas estrategias pueden afectar a las capacidades de los estudiantes. También ofrece información útil para la gestión de la calidad en los programas educativos, ya que permite detectar necesidades de reforzamiento de los nuevos estudiantes y ayuda a mejorar el contenido y el diseño de los futuros programas académicos.

Oele es un sistema utilizado para establecer las correspondencias entre las respuestas dadas por los estudiantes con respecto a lo definido en un mapa conceptual u ontología del dominio del conocimiento, diseñado y examinado por especialistas en el tema. En el proyecto *Ontology-Based E-Assessment for Accounting* (Litherland, Carmichael & Martínez-García, 2013) se ha desarrollado y explorado el potencial de un sistema de e-evaluación basado en una ontología que se nutre del potencial de las tecnologías semánticas emergentes. Este sistema ofrece un entorno de evaluación *online* capaz de valorar respuestas de texto libre de los estudiantes a preguntas basadas en conceptos, en lugar de a preguntas de múltiple elección basadas en hechos.

Valorar los escritos de los estudiantes suele ser una tarea que puede consumir grandes períodos de tiempo. En (Hea, Huib & Quanc, 2009) se describe cómo la evaluación asistida por ordenador puede ayudar a los profesores a mejorar este proceso y hacerlo de una manera más efectiva. En (Csapó, Ainley, Bennett, Latour & Law, 2012) se muestran algunos casos especiales donde las tecnologías de la información han ofrecido diversos avances significativos, como por ejemplo, en la evaluación de estudiantes con necesidades especiales, evaluación de capacidades colaborativas y los logros de los grupos de alumnos, etc.

3. Evaluación sostenible de experiencias de aprendizaje

Nuestra propuesta consiste en un marco de trabajo para la evaluación sostenible de experiencias de aprendizaje, el cual se compone de una serie de criterios, indicadores y fuentes de datos. Esta sección incluye además un ejemplo de cómo puede ser aplicado este marco de trabajo para construir una arquitectura software para la recomendación de evaluaciones.

3.1. Criterios de evaluación

Los criterios a evaluar en el contexto de una experiencia de aprendizaje basada en proyectos pueden ser muy numerosos y cada uno de ellos puede ser evaluado para comprobar diferentes resultados de aprendizaje. Así pues, nuestro método de evaluación se ha limitado a un conjunto reducido de criterios que permiten evaluar varias competencias:

- *C1: Asignación equilibrada de tareas a los roles del proyecto.* Este criterio permite evaluar las competencias genéricas e instrumentales de *capacidad de organización y planificación*, y la de *resolución de problemas*, así como la competencia general y personal de *trabajo en equipo*.
- *C2: Uso adecuado de la forja de software.* Este criterio permite medir las competencias específicas de *conocimiento de las herramientas de desarrollo web*.
- *C3: Cumplimiento de los hitos y tareas en tiempo.* Este criterio permite medir las competencias genéricas e instrumentales de *capacidad de organización y planificación*.

3.2. Fuentes de datos e indicadores

El entorno virtual de aprendizaje es la fuente principal de los datos para el análisis de los indicadores. Asimismo, el propio VLE es un entorno de aprendizaje personal (Liber & Johnson, 2008) compuesto por diferentes herramientas ofrecidas por las forjas de software. En nuestro caso, nuestra propuesta hace uso de los siguientes sistemas para dar soporte a las capacidades esperadas en las experiencias PBL:

- *Assembla Tickets*, para la planificación y seguimiento de las tareas de desarrollo. Gracias a su interfaz programática (API), la herramienta ofrece información interesante sobre la distribución del trabajo en cada uno de los proyectos registrados. Asimismo, la herramienta es compatible con cualquiera de los entornos de desarrollo utilizados por los alumnos.
- *Subversion y Git*, a elección de cada equipo de proyecto, que permite gestionar el repositorio de control de versiones del código fuente. Estas herramientas registran cada uno de los cambios sobre los ficheros del proyecto, contemplando qué se modifica y quién lo hace.

- *MediaWiki*, que permite almacenar y gestionar los entregables documentales de los proyectos software, como los requisitos de software, casos de uso, etc. Se trata de la herramienta de edición colaborativa más popular y alrededor de la misma existen varias utilidades que permiten extraer información de interés.

Los siguientes indicadores fueron diseñados para observar las evidencias a favor de los criterios de evaluación, para cada aplicación que actúa como fuente de datos.

Assembla Tickets

- Retraso en la compleción de la última tarea (*ticket*) en cada hito (*milestone*) del proyecto.
- Número de tareas abiertas, asignadas y resueltas para cada usuario y equipo.
- Intervalo de tiempo entre la creación y el cierre de cada tarea.
- Diferencia entre el número de tareas abiertas y tareas cerradas o resueltas.

MediaWiki

- Número de ediciones de texto y cantidad de texto aportado.
- Clasificación de cada miembro del equipo como contribuidor en la wiki.
- Reparto en la distribución del número de contribuciones a lo largo de la vida del wiki.

Subversion y Git

- Número de aportes (*commits*) de cada individuo y del total del equipo.
- Diferencia de tiempo entre cada par de aportes.

3.3. Modelo de datos

En la figura 1 se presenta el modelo conceptual diseñado para nuestro marco de trabajo. El elemento fundamental de este modelo es el proyecto (*project*), el cual a su vez se compone de hitos (*milestone*), tareas (*ticket*), usuarios (*user*) y resultados (*metric results*). El elemento conjunto de datos (*dataset*) hace posible analizar y comprobar el progreso de un determinado proyecto a lo largo del tiempo desde las forjas de *software*. Cada uno de los conjuntos de datos se asocia a las forjas (*forge*) de *software* en un determinado instante de tiempo.

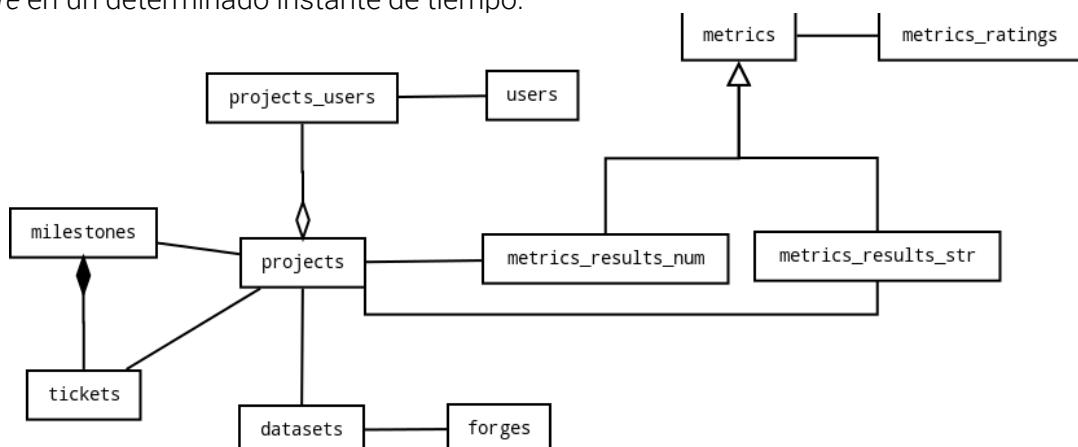


Figura 1. Modelo conceptual de datos

En la tabla 1 se recogen los principales atributos de interés de las tareas y de los hitos. Existen dos tipos fundamentales de métricas: aquellas representadas de forma numérica y aquellas basadas en texto.

Table	Metric
Tickets	Creation date
	Completion date
	Author
	Owner
	Milestone to which it belongs
	State (opened, closed, etc.)
Milestones	Creation Date
	Due date
	Completion date
	User creator

Tabla 1. Atributos de interés de las tareas e hitos

En la tabla 2 se incluyen las principales métricas que utilizamos en nuestro marco de trabajo, mientras que en la tabla 3 se indican cuáles de esas métricas se usan para evaluar cada criterio.

Type	Name	Description
Numeric	Deadline fulfillment	Retraso medio en la realización de los hitos
	Teamwork balance	Desviación típica del número de tareas asignadas a cada miembro del equipo
	Planning creations	Máximo número de días entre creaciones consecutivas de tareas
	Planning updates	Máximo número de días entre actualizaciones consecutivas de tareas
	Version control	Número máximo de días entre <i>commits</i> consecutivos
String	Resolutive capability	Usuario que ha cerrado más tareas (<i>tickets</i>)
	Leadership	Usuario que ha creado más tareas (<i>tickets</i>)

Tabla 2. Clasificación de métricas

Criteria	Metrics
C1	Teamwork balance
	Leadership
	Resolutive capability
C2	Planning creations
	Planning updates
	Version Control
C3	Deadline fulfillment

Tabla 3. Métricas utilizadas para cada criterio

Resulta de interés describir el elemento *metric_rating*, el cual representa las correspondencias entre el rango de valores de cada métrica y la puntuación o calificación (en una escala de 0 a 10) asignada por el supervisor. Por ejemplo, si el valor de la métrica *Deadline fulfillment* (que representa al retraso promedio para completar los hitos de entrega) está por debajo de 0, implica que el equipo de trabajo acaba con antelación lo prescrito para cada hito, con lo cual se le podría asignar una calificación de 10. A partir de aquí, por cada día promedio de retraso se le aminora la calificación en una unidad, esto es, 9 puntos para un promedio de 1 día de retraso, 8 puntos para un promedio de 2 días, etc.

3.4. Análisis de ejemplos de evaluación

Algunos de los indicadores que ofrecen las herramientas anteriormente descritas fueron utilizados como fuentes de evidencias para los criterios de evaluación. Por ejemplo, el criterio *C1-Asignación equilibrada de tareas a los roles del proyecto* fue observado y analizado principalmente desde dos fuentes principales: el número de tareas registradas en *Assembla Tickets* y el número de ediciones de texto llevadas a cabo sobre los entregables alojados en *Mediawiki*. Los primeros pueden ser obtenidos directamente mediante la observación de los informes generados desde la web de la propia herramienta, mientras que para calcular los segundos se requiere de una herramienta externa para ello, como *StatMediaWiki* (Palomo-Duarte, Dodero, Medina-Bulo, Rodríguez-Posada & Ruiz-Rube, 2014). Puede observarse un informe completo de las tareas llevadas a cabo (<http://bit.ly/tckblnc>) y de la distribución de las ediciones en la wiki (<http://bit.ly/wkiblnc>) por parte de los miembros de los equipos de proyecto. Estos informes estuvieron disponibles públicamente para los estudiantes durante el desarrollo del curso.

La métrica *Teamwork balance* puede ser calculada mediante el conteo del número de tareas asignadas a cada uno de los miembros de los grupos de trabajo. Por ejemplo, en la tabla 4 se muestra la distribución de tareas en un cierto equipo, donde la desviación estándar es 9, lo que muestra una distribución de tareas poco balanceada.

Student	Tickets assigned
Student1	9
Student2	4
Student3	9
Student4	27

Tabla 4. Ejemplo de distribución de tareas

Esta métrica puede ser complementada con otras procedentes de la wiki. La figura 2 muestra un diagrama con la distribución de la carga de trabajo de un determinado grupo. Puede observarse en primer lugar que el estudiante de color púrpura parece ser el líder del grupo, ya que el 80% del contenido

de la wiki es creado por ese miembro del equipo. Después de 10 días, el estudiante representado por el color azul fue el único que había ayudado al de color púrpura, ya que añadió bastante contenido, siendo responsable de la mayor parte del contenido de la wiki. Inmediatamente, los estudiantes de color verde y rojo hicieron sus primeras contribuciones. Al final (véase tabla 5), el estudiante de color azul sigue siendo el máximo contribuidor de su grupo con un 65% de las contribuciones, mientras que el resto oscila entre el 20% y el 7% de las contribuciones.

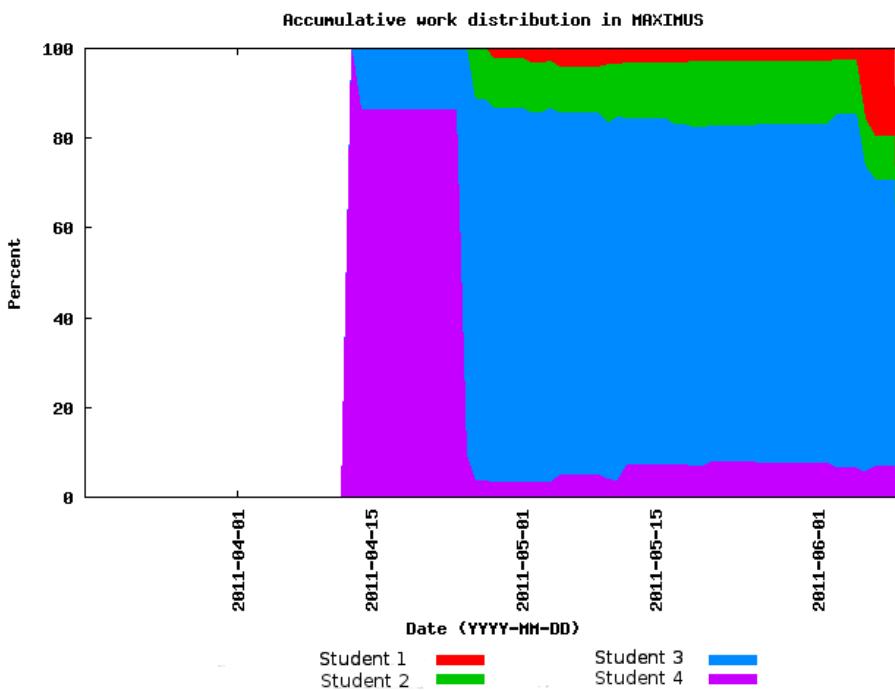


Figura 2. Distribución de la carga de trabajo

Adicionalmente los indicadores basados en las tareas asignadas o resueltas empleados para evaluar el criterio *C1-Capacidad de organización y planificación*, puede proporcionar evidencias relativas a otras competencias. Específicamente, pueden ser útiles para evaluar la *capacidad de resolución de problemas*, ya que los *tickets* no solo son un reflejo de las tareas planificadas, sino que también lo son de tareas emergentes para la resolución de incidencias.

Estas tareas son generadas durante la ejecución de los proyectos y pueden ser asignadas a los desarrolladores que desempeñan un rol diferente al inicialmente planificado, lo que se conoce como *Backup behavior*. Sin embargo, ya que las herramientas empleadas no definen los roles asignados a cada miembro de los equipos, no es factible obtener de forma sencilla indicadores que ofrezcan las evidencias de este comportamiento.

Con el objetivo de medir la actividad en el sistema de control de versiones, se ha empleado la herramienta *CVSAnaly* (Robles, Koch & González-Barahona, 2004), con la cual se puede observar el número de total de *commits* realizados y por consecuencia, el nivel de uso de este tipo de herramientas. Esta evaluación puede ser acompañada por otro indicador que mide el total de tareas creadas en

Student	Edits	Edits rate	Bytes	Bytes rate
Blue	101	51,5%	44.998	65,7%
Red	59	30,1%	13.945	20,4%
Purple	21	10,7%	4.953	7,2%
Green	15	7,7%	6.875	10,0%

Tabla 5. Contribuciones del usuario a la wiki al final del curso

la herramienta *Assemba Tickets*. La combinación de ambos indicadores permite evaluar el uso de la forja de software (criterio C2) a nivel de equipo. Además, es importante medir la frecuencia de esta actividad. Normalmente el objetivo es que el trabajo sea continuo durante el tiempo, sin grandes irregularidades. Las métricas *Planning creations*, *Planning updates* y *Version control* ofrecen información sobre esto. Observando la tabla 6 puede concluirse el valor de las dos primeras métricas. Los mayores periodos de inactividad son 15 y 16 días, respectivamente. La misma idea se aplica en la métrica de *Version control*, de tal modo que el periodo más largo de inactividad con respecto al control de versiones es de 5 días, tal y como se puede observar en la tabla 7.

Ticket	Creation	Elapsed days	Update	Elapsed days
1	12/01/2012	-	15/01/2012	-
2	17/01/2012	5	24/01/2012	9
3	01/02/2012	15	02/02/2012	9
4	11/02/2012	10	18/02/2012	16

Tabla 6. Creaciones y actualizaciones de tareas

Commit	Date	Elapsed days
1	14/01/2012	-
2	14/01/2012	0
3	15/01/2012	1
4	17/01/2012	2
5	17/01/2012	0
6	17/01/2012	0
7	22/01/2012	5

Tabla 7. Registro de commits en el control de versiones

Todas esas métricas se utilizan en un contexto en el que se supone que los desarrolladores hacen un uso adecuado de las herramientas. Para lograr este objetivo, en clase se enseñó las siguientes pautas a los alumnos:

- Todos los cambios sobre el código fuente debe reflejarse en la forja. Nadie debe desarrollar código al margen.
- Los errores detectados durante el proyecto deben registrarse en el sistema de gestión de tickets.

- La planificación del proyecto debe realizarse con la ayuda de la herramienta para gestión de hitos de *assembla*.
- Los desarrolladores deben actualizar los cambios relativos a las tareas gestionadas en el sistema de gestión de tickets.

El criterio *C3* relativo al cumplimiento de los hitos en tiempo fue analizado utilizando la herramienta *Assembla Milestones*, la cual ofrece una serie de informes sobre el cumplimiento de las fechas programadas en los hitos planificados y el esfuerzo (estimado vs. incurrido) para alcanzar ese hito. En <http://bit.ly/mlstons> puede observarse el informe de los hitos de un proyecto.

La métrica *deadline fulfillment* puede ser calculada utilizando la información que aparece en la tabla 8, donde pueden comprobarse las fechas límite de cada hito de un proyecto determinado y la fecha de cierre de la última tarea asignada a ese hito. La métrica mide el número medio de días de retraso, siendo en este caso de 9 días.

Milestone	Due date	Last ticket closing	Delay
Sprint 0	18/04/2012	20/04/2012	2
Sprint 1	02/05/2012	02/05/2012	0
Sprint 2	09/05/2012	22/05/2012	13
Sprint 3	16/05/2012	22/05/2012	6
Sprint 4	23/05/2012	11/06/2012	19
Sprint 5	30/05/2012	03/06/2012	4
Sprint 6	13/06/2012	04/06/2012	-9

Tabla 8. Cumplimiento de hitos de entrega

3.5. Arquitectura

Con el objetivo de calcular las métricas descritas anteriormente implementamos una solución de integración de datos basada en el patrón *Extraction, Transformation and Load* (ETL) (Vassiliadis, 2009). En la literatura se puede encontrar algún trabajo relacionado como el utilizado en la herramienta *Spago4Q* (Colombo *et al.*, 2008) pero que está orientado al entorno industrial, y no al académico. La solución implementada se basa en la suite de análisis *Pentaho Business Intelligence* (Bouman & Van Dongen, 2009), específicamente haciendo uso de la herramienta *Pentaho Data Integration*.

El proceso de integración utiliza el API de la forja de *Assembla* y de los diferentes repositorios de código fuente para poder extraer los datos necesarios para analizar los proyectos desarrollados por los alumnos. En la figura 3 se puede observar, con un alto nivel de abstracción, cada uno de los pasos del proceso de integración.

El proceso comienza obteniendo la lista de proyectos a analizar para la evaluación. Después de eso,

el proceso se conecta a las forja utilizando la correspondiente API para descargar los metadatos del proyecto en formato XML. Luego, esos datos son transformados en un esquema común de proyectos de software independiente de la forja. De esta manera, las tareas, los hitos y el resto de información de los proyectos serán finalmente almacenados en una base de datos relacional, denominada *Abreforjas*, para su posterior procesamiento. En el anexo al artículo se incluye un extracto de código en formato RDF/Turtle correspondiente a la formalización de la base de datos.

En una segunda etapa de procesamiento todos los ficheros de código fuente almacenados en los repositorios de código de cada proyecto son descargados, haciendo uso de un script en *Python*. Luego, se emplea la herramienta *CVSAnaly*, un *software* para calcular métricas de revisiones de código fuente, con el cual se colecciona información sobre los *commits* registrados, el autor de cada uno de ellos, etc. Este proceso almacena los resultados en otra base de datos relacional.

Finalmente, una vez que los datos relativos al control de versiones y a la gestión de tareas son almacenados localmente, un proceso posterior se encarga de fusionar toda esa información en una única vista de datos global. Posteriormente, se llevan a cabo una serie de operaciones para calcular las métricas que permitan evaluar el desarrollo de las competencias planteadas. Estos resultados fueron analizados gráficamente utilizando algunas operaciones estadísticas ofrecidas por el paquete estadístico R (<http://www.r-project.org>).

Al margen de esto, la información recopilada por la herramienta *StatMediaWiki* fue tenida en cuenta para el análisis, aunque su tratamiento no fue integrado en la arquitectura de integración.

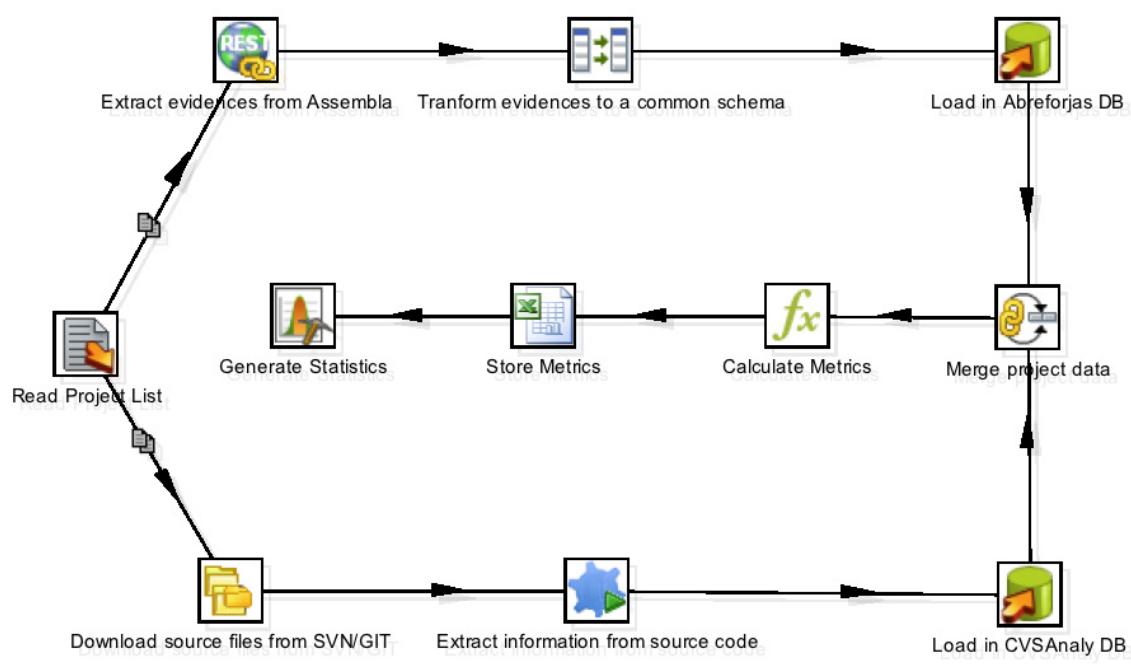


Figura 3. Proceso de integración

4. Caso de estudio

El caso de estudio descrito a continuación se centra en el contexto de una asignatura de *Ingeniería Web* de la titulación universitaria de Ingeniería Informática. Las competencias más significativas desarrolladas en esta asignatura son la capacidad de organización y planificación, y el trabajo en equipo (genéricas), el conocimiento de los métodos, técnicas y herramientas para el desarrollo de aplicaciones web (cognitivas) y su aplicación en el ciclo completo de ingeniería de una aplicación web (procedimental), además de la motivación para la calidad en el desarrollo de *software* y el aprendizaje de nuevos métodos y herramientas (actitudinal) (Hernández-Leo *et al.*, 2012).

El método utilizado se basa en el aprendizaje basado en proyectos y colaborativo. Durante la primera parte del curso (7 semanas), los estudiantes desarrollaron una serie de ejercicios, focalizados en el aprendizaje de nuevos métodos y *frameworks*, sobre una wiki. En la segunda parte (8 semanas), los alumnos trabajaron en equipos pequeños para desarrollar una aplicación web siguiendo todas las fases del ciclo de vida de ingeniería del *software*. Con idea de hacer el trabajo de los estudiantes más sencillo, se desarrolló una rúbrica con las habilidades a evaluar. Esta rúbrica fue compartida con los estudiantes para permitirles que se familiaricen con las habilidades a evaluar y que, por tanto, tengan que potenciar.

4.1. Preparación

Durante las sesiones de prácticas, a los estudiantes se les enseñó cómo evaluar las habilidades utilizando la rúbrica, usando como indicadores las métricas disponibles en la forja de *software* y en los instrumentos de análisis de los entregables textuales desplegados sobre la wiki. Debido a que los criterios de evaluación son numerosos (32 criterios), se encuentran publicados en el sitio web¹ proporcionado junto con este caso de estudio. Cada uno de ellos puede ser evaluado mediante diferentes resultados de aprendizaje. Las sesiones formativas de evaluación fueron reducidas a un conjunto limitado de criterios (C1-C3) del enfoque de evaluación sostenible descrito anteriormente.

Los entregables textuales de los proyectos son alojados en la wiki de los proyectos, mientras que el código fuente se almacena en el sistema de control de versiones vinculado con a la forja de *software*. A los estudiantes se les pidió que realizaran evaluaciones formativas de sus propios entregables y de los de otros grupos. Finalmente el profesor hizo una evaluación global de los proyectos, cuyos resultados fueron publicados y discutidos en co-evaluación con los estudiantes.

¹ http://spi-fm.uca.es/spdef/#cs_susteval

4.2. Resultados

El curso PBL lo realizó un total de 4 equipos en el año 2009, y por una media de 6 o 7 equipos en los siguientes cursos académicos, desde 2010 hasta 2012. Cada equipo se compone de 4 integrantes. En las figuras 4, 5 y 6 se muestran la evolución de la puntuación obtenida para los criterios C1-*Asignación equilibrada de tareas a los roles del proyecto*, C2-*Uso adecuado de las herramientas de las herramientas de la forja de software* y C3-*Cumplimiento de los hitos y tareas en tiempo*.

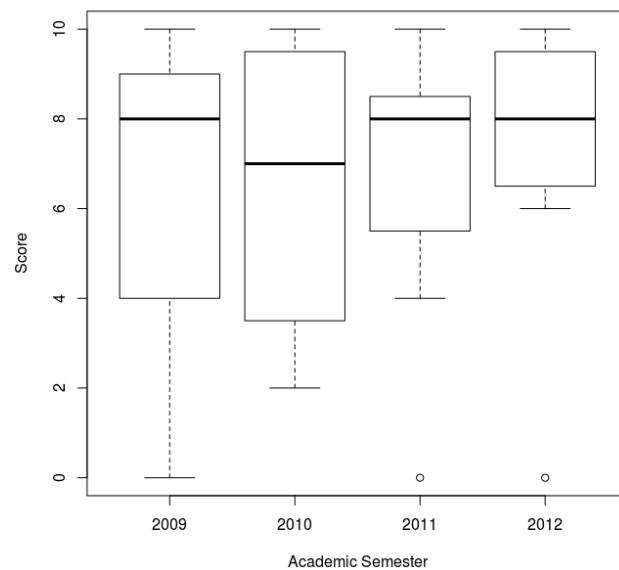


Figura 4. Evaluación del criterio C1

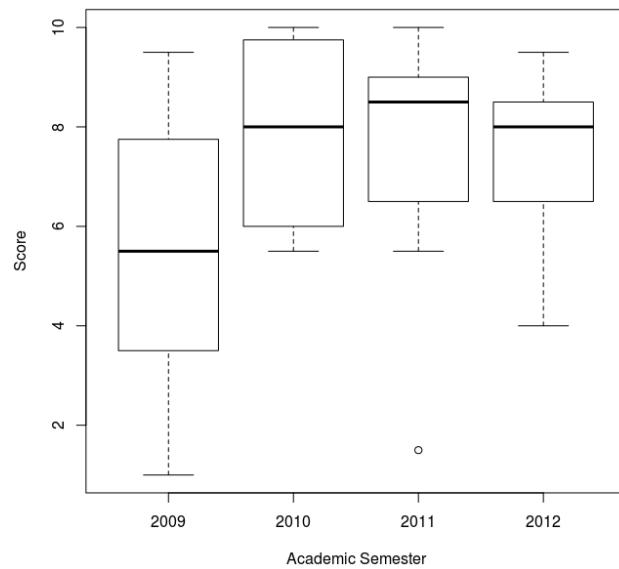


Figura 5. Evaluación del criterio C2

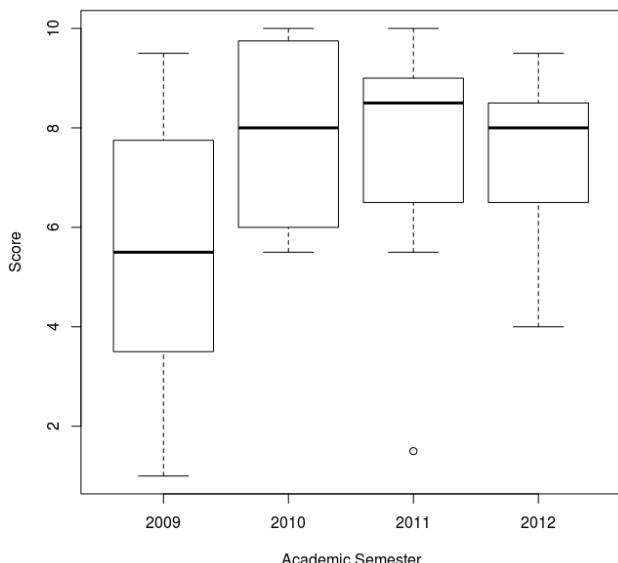


Figura 6. Evaluación del criterio C3

Hasta el momento, los resultados indican aparentemente que no hay evidencias significativas en detrimento de nuestra propuesta de evaluación sostenible mediante evaluación global de los estudiantes. Incluyendo un grupo que no siga esta metodología y exponiendo a ambos grupos el mismo examen, podría proporcionar un contraste más exacto. Con el objetivo de asegurar la reproducibilidad de los experimentos, se ha publicado un sitio web con toda la información relacionada con el caso de estudio, incluyendo la documentación del proceso de integración, el código fuente de los proyectos analizados y las métricas obtenidas para cada competencia, entre otros datos.

5. Conclusiones

En este trabajo se describe una propuesta de e-evaluación que, a diferencia de otras anteriores, se focaliza en la evaluación de experiencias de aprendizaje durante el desarrollo de proyectos *software*. Nuestro enfoque también se basa en la reutilización de la información obtenida a partir de herramientas de terceros que sirven de ayuda al proceso de desarrollo del *software*. La información recopilada por dichas herramientas permite la obtención e integración de métricas de evaluación de una manera no intrusiva.

Una forja de código es una herramienta adecuada para el soporte del desarrollo colaborativo de proyectos *software*. Sin embargo, una evaluación detallada utilizando la información recolectada por dicha forja convierte la evaluación en un proceso menos sostenible, aun cuando los miembros de los diferentes equipos se involucran en procedimientos como la autoevaluación o la evaluación por pares. En este trabajo proponemos una arquitectura para la extracción de diferentes indicadores que pueden facilitar este proceso. Dicha arquitectura está basada en diferentes sistemas de fuentes abiertas interconectados utilizando *Pentaho Data Integration* y otras aplicaciones como *Assembia* y

CVSAnalY. Para el análisis de la documentación presente en *MediaWiki* se utilizó *StatMediaWiki*. Se han utilizado modelos de datos sencillos y representaciones de datos simples. Una alternativa podría haber sido utilizar *linked data* (Bizer & Berners-Lee, 2009) para interpretar analíticas de aprendizaje. *Linked data*, sin embargo, se basa en la existencia de un número suficiente de conjuntos de datos (D'Aquin, 2012) que, desafortunadamente, no están aún disponibles. Un paso más hacia la sencillez puede observarse en la comunidad de *Linked Open Data*, que ha animado a los desarrolladores a utilizar las mismas URIs para identificar entidades en diferentes repositorios de datos (Karger, 2013). Por el momento, nos hemos apartado de los modelos de *Linked Data* para establecer una integración sencilla de datos abiertos que permita obtener los datos necesarios.

La experiencia aporta evidencias a favor del desarrollo de prácticas de evaluación sostenible en experiencias PBL. Dichas prácticas se basan en la familiarización de los miembros del equipo con un conjunto de indicadores de actividad que pueden ser utilizados para la recomendación de evaluaciones. Las notas obtenidas de esta manera son similares a las correspondientes a los cursos anteriores, estando estas sustentadas en las evidencias observadas y habiendo sido calculadas con menor esfuerzo. Además, las evaluaciones pueden ser formativas para los miembros de los equipos, dado que pueden incrementar su capacidad de pensamiento crítico. De cualquier modo, es necesario un mayor número de datos experimentales para hacer estas conclusiones más sólidas.

En ausencia de una estrategia de datos abiertos, los consumidores de fuentes de datos deben desarrollar *scrapers* que extraigan la información de las fuentes de datos no estructuradas, o bien extraer y transformar la información de cada elemento del conjunto de datos de manera manual. El enfoque de datos abiertos que afrontamos en este trabajo es el precursor necesario para conseguir analíticas de aprendizaje más sofisticadas que podrán desarrollarse en un futuro. Por ejemplo, el uso de *linked data* permitiría integrar recomendaciones formativas de evaluación en forma de proalimentación bajo herramientas externas de evaluación basada en rúbricas. Por otro lado, se pretende simplificar la definición de las diferentes métricas y procesos de evaluación como trabajo parte del futuro, extendiendo un lenguaje de dominio específico para propósitos de evaluación (Dodero, Ruiz-Rube, Palomo-Duarte & Cabot, 2012).

6. Anexo

Un extracto del vocabulario RDF utilizado por la base de datos *Abreforjas* y que define las características comunes de las herramientas de gestión de tareas como *Assembla*, *Jira*, *Redmine* o *Trac* se incluye a continuación.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix owl: <http://www.w3.org/2002/07/owl#> .
```

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix itm: <http://spi-fm.uca.es/spdef/models/genericTools/
itm/1.0#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<http://spi-fm.uca.es/spdef/models/genericTools/itm/1.0> a
owl:Ontology;
    dc:title "Issue Tracking tool Model";
    dc:modified "2013-12-15"^^xsd:date;
    foaf:page <http://spi-fm.uca.es/spdef/models/genericTools/
itm/1.0.html> .
itm:Role a rdfs:Class, owl:Class .
itm:Version a rdfs:Class, owl:Class;
    rdfs:subClassOf <http://usefulinc.com/ns/doap#Version> .
itm:User a rdfs:Class, owl:Class;
    rdfs:subClassOf foaf:Person .
itm:Member a rdfs:Class, owl:Class .
itm:Tracker a rdfs:Class, owl:Class .
itm:IssueDependency a rdfs:Class, owl:Class .
itm:Issue a rdfs:Class, owl:Class .
itm:IssueCategory a rdfs:Class, owl:Class .
itm:IssueTrackingDatabase a rdfs:Class, owl:Class .
itm:Project a rdfs:Class, owl:Class;
    rdfs:subClassOf <http://usefulinc.com/ns/doap#Project> .
itm:issues a rdf:Property;
    rdfs:domain itm:Version;
    rdfs:range itm:Issue .
itm:language a rdf:Property, owl:DatatypeProperty;
    rdfs:domain itm:User;
    rdfs:range xsd:string .
itm:login a rdf:Property, owl:DatatypeProperty;
    rdfs:domain itm:User;
    rdfs:range xsd:string .
itm:tracker a rdf:Property;
    rdfs:domain itm:Issue;
    rdfs:range itm:Tracker .
itm:dueDate a rdf:Property, owl:DatatypeProperty;
    rdfs:domain itm:Issue;
    rdfs:range xsd:date .
itm:priority a rdf:Property;
    rdfs:domain itm:Issue;
    rdfs:range skos:Concept .
itm:dependencies a rdf:Property;
    rdfs:domain itm:Issue;
    rdfs:range itm:IssueDependency .
itm:status a rdf:Property;
    rdfs:domain itm:Version,itm:Issue;
    rdfs:range skos:Concept .
```

```
itm:estimatedHours a rdf:Property, owl:DatatypeProperty;
  rdfs:domain itm:Issue;
  rdfs:range xsd:float .
itm:name a rdf:Property, owl:DatatypeProperty;
  rdfs:domain
ject,itm:Version,itm:Issue,itm:Tracker,itm:IssueCategory,itm:Role;
  rdfs:range xsd:string;
  rdfs:subPropertyOf dc:title .
itm:roles a rdf:Property;
  rdfs:domain itm:IssueTrackingDatabase;
  rdfs:range itm:Role .
itm:user a rdf:Property;
  rdfs:domain itm:Member;
  rdfs:range itm:User .
itm:issueCategories a rdf:Property;
  rdfs:domain itm:Project;
  rdfs:range itm:IssueCategory .
itm:completedDate a rdf:Property, owl:DatatypeProperty;
  rdfs:domain itm:Version, itm:Issue;
  rdfs:range xsd:date .
itm:users a rdf:Property;
  rdfs:domain itm:IssueTrackingDatabase;
  rdfs:range itm:User .
itm:permissions a rdf:Property, owl:DatatypeProperty;
  rdfs:domain itm:Role;
  rdfs:range xsd:string .
itm:trackers a rdf:Property;
  rdfs:domain itm:IssueTrackingDatabase;
  rdfs:range itm:Tracker .
itm:elapsedHours a rdf:Property, owl:DatatypeProperty;
  rdfs:domain itm:Issue;
  rdfs:range xsd:float .
itm:description a rdf:Property, owl:DatatypeProperty;
  rdfs:domain itm:Project,itm:Version,itm:Issue;
  rdfs:range xsd:string;
  rdfs:subPropertyOf dc:description .
itm:projects a rdf:Property;
  rdfs:domain itm:IssueTrackingDatabase;
  rdfs:range itm:Project .
itm:versions a rdf:Property;
  rdfs:domain itm:Project;
  rdfs:range itm:Version .
itm:type a rdf:Property;
  rdfs:domain itm:IssueDependency;
  rdfs:range skos:Concept .
itm:role a rdf:Property;
  rdfs:domain itm:Member;
  rdfs:range itm:Role .
itm:category a rdf:Property;
  rdfs:domain itm:Issue;
```

```
rdfs:range itm:IssueCategory .  
itm:members a rdf:Property;  
    rdfs:domain itm:Project;  
    rdfs:range itm:Member .  
itm:owner a rdf:Property;  
    rdfs:domain itm:Issue;  
    rdfs:range itm:Member .  
itm:dependentTask a rdf:Property;  
    rdfs:domain itm:IssueDependency;  
    rdfs:range itm:Issue .  
itm:responsible a rdf:Property;  
    rdfs:domain itm:Issue;  
    rdfs:range itm:Member .  
itm:doneRatio a rdf:Property, owl:DatatypeProperty;  
    rdfs:domain itm:Issue;  
    rdfs:range xsd:float .
```

7. Agradecimientos

Este trabajo ha sido financiado por medio del programa PAIDI de la Junta de Andalucía en el contexto del proyecto *Asceta* (P09-TIC-5230) y mediante el programa de Proyectos de Innovación Educativa Universitaria del Personal Docente e Investigador de la Universidad de Cádiz en el contexto del proyecto *Evaluación sostenible de actividades complejas de aprendizaje en forjas de desarrollo de software* (PI2_12_030).

8. Referencias

Achcaoucaou, F., Guitart-Tarrés, L., Miravitles-Matamoros, P., Núñez-Carballosa, A., Bernardo, M. & Bikfalvi, A. (2014). Competence assessment in higher education: A dynamic approach. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 454-467. doi: <http://dx.doi.org/10.1002/hfm.20394>

Amelung, M., Krieger, K. & Rosner, D. (2011). E-Assessment as a Service. *Learning Technologies, IEEE Transactions on*, 4(2), 162-174. doi:<http://dx.doi.org/10.1109/TLT.2010.24>

Barootchi, N. & Keshavarz, M. H. (2002). Assessment of achievement through portfolios and teacher-made tests. *Educational Research*, 44(3), 279-288. doi:<http://dx.doi.org/10.1080/00131880210135313>

Bizer, C. H. & Berners-Lee, T. (2009). Linked data-the story so far. *Semantic Services, Interoperability and*

Boud, D. (2000). Sustainable assessment: rethinking assessment for the learning society. *Studies in continuing education*, 22(2), 151-167. doi:<http://dx.doi.org/10.1080/713695728>

Boud, D. (2007). Reframing assessment as if learning were important. *Rethinking assessment in higher education: Learning for the longer term*, 14-25.

Bouman, R. & Van Dongen, J. (2009). *Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL*. Wiley.

Colombo, A., Damiani, E., Frati, F., Oltolina, S., Reed, K. & Ruffatti, G. (2008). The use of a meta-model to support multi-project process measurement. *15th Asia-Pacific Software Engineering Conference, 2008. APSEC'08* (pp. 503-510). IEEE. doi:<http://dx.doi.org/10.1109/APSEC.2008.55>

Csapó, B., Ainley, J., Bennett, R. E., Latour, T. & Law, N. (2012). Technological issues for computer-based assessment. *Assessment and teaching of 21st century skills*, 143-230. doi:http://dx.doi.org/10.1007/978-94-007-2324-5_4

D'Aquin, M. (2012). Putting linked data to use in a large higher-education organisation. *Proceedings of the Interacting with Linked Data (ILD) workshop at Extended Semantic Web Conference (ESWC)*. Heraklion, Crete.

Davies, S. (2010). *Effective Assessment in a Digital Age*.

Dodero, J. M., Ruiz-Rube, I., Palomo-Duarte, M. & Cabot, J. (2012). Model-driven learning design. *Journal of Research and Practice in Information Technology*, 44(3), 267-288.

Falchikov, N. (2013). *Improving assessment through student involvement: Practical solutions for aiding learning in higher and further education*. Routledge.

Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, 4(5-6), 304-317. doi:<http://dx.doi.org/10.1504/IJTEL.2012.051816>

Gala-Pérez, S. J. (2013). Intensive metrics for the study of the evolution of open source projects. *Master's thesis, Universidad Rey Juan Carlos*. <http://hdl.handle.net/10115/11518>.

Gil, C., Montoya, M. G., Herrada, R. I., Banos, R., Montoya, F. G. & Manzano-Agugliaro, F. (2011). Cooperative learning and electronic group portfolio: tutoring tools, development of competences and assessment. *International Journal of Learning Technology*, 6(1), 46-61. doi: <http://dx.doi.org/10.1504/IJLT.2011.040149>

Hea, Y., Huib, S. C. & Quanc, T. (2009). Automatic summary assessment for intelligent tutoring systems. *Computers & Education*, 53(3), 890-899. doi:<http://dx.doi.org/10.1016/j.compedu.2009.05.008>

Hernández-Leo, D., Moreno, V., Dodero, J., Pardo, A., Romero-Ternero, M., Dimitriadis, Y. & Asensio-Pérez, J. I. (2012). Aplicación de Recomendaciones para la alineación de Competencias, Metodología y evaluación en asignaturas de Ingeniería Telemática, Informática y Electrónica. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 7(1), 13-20.

Johnson, D. H. (2013). Teaching a "MOOC": Experiences from the front line. *Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), 2013 IEEE*, 268-272. doi:<http://dx.doi.org/10.1109/DSP-SPE.2013.6642602>

Karger, D. (2013). Standards opportunities around data-bearing Web pages. *Philosophical Transactions of the Royal Society A*, 371(1987), 20120381. doi:<http://dx.doi.org/10.1098/rsta.2012.0381>

Klenowski, V., Askew, S. & Carnell, E. (2006). Portfolios for learning, assessment and professional development in higher education. *Assessment & Evaluation in Higher Education*, 31(3), 267-286. doi:<http://dx.doi.org/10.1080/02602930500352816>

Liber, O. & Johnson, M. (2008). Personal learning environments. *Interactive Learning Environments*, 16(1), 1-2. doi:<http://dx.doi.org/10.1080/10494820701772645>

Lim, C. P., Lai, Y. C. & Ng, E. M. (2011). Using wikis to develop student teachers' learning, teaching, and assessment capabilities. *The Internet and Higher Education*, 14(1), 15-26. doi:<http://dx.doi.org/10.1016/j.iheduc.2010.06.001>

Litherland, K., Carmichael, P. & Martínez-García, A. (2013). Ontology-based e-Assessment for Accounting Education. *Accounting Education*, 22(5), 498-501. doi:<http://dx.doi.org/10.1080/09639284.2013.824198>

Lugton, M. (2012). *What is a MOOC? What are the different types of MOOC? xMOOCs and cMOOCs*. Reflections.

Markham, T. & Larmer, J. (2003). *Project Based Learning Handbook: A Guide to Standards-Focused Project Based Learning for Middle and High School Teachers*. Buck Institute for Education.

Martín-Cuadrado, A. M., López-González, M. A. & García-Arce, A. (2013). Innovation Network: Videoconferencing as a Resource in Teaching Support and Autonomous Learning. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 8(3), 119-125. doi:<http://dx.doi.org/10.1109/RITA.2013.2273111>

Masip-Álvarez, A., Hervada-Sala, C., Pàmies-Gómez, T., Arias-Pujol, A., Jaen-Fernandez, C., Rodriguez-Sorigue, C., Montferrer-Linan, R. (2013). Self-video recording for the integration and assessment of generic competencies. *Global Engineering Education Conference (EDUCON), 2013 IEEE*, 436-441. doi:<http://dx.doi.org/10.1109/EduCon.2013.6530142>

Mor, Y. & Koshinen, T. (2013). MOOC y más allá. *eLearning Papers*, 33, 1-7.

Munkhchimeg, B. & Baigaltugs, S. (2013). Control possibility of students' learning process through using Learning Management System. *Ifost IEEE*, 2, 395-399. doi:<http://dx.doi.org/10.1109/IFOST.2013.6616920>

Palomares, A. (2011). The educational model at university and the use of new methodologies for teaching, learning and assessment. *Revista de Educación*, 355(Mayo-Agosto 2011), 591-604.

Palomo-Duarte, M., Dodero, J. M., Medina-Bulo, I., Rodríguez-Posada, E. J. & Ruiz-Rube, I. (2014). Assessment of collaborative learning experiences by graphical analysis of wiki contributions. *Interactive Learning Environments*, 22(4), 444-466. doi:<http://dx.doi.org/10.1080/10494820.2012.680969>

Piedra, N., Chicaiza, J., López, J., Romero, A. & Tovar, E. (2010). Measuring collaboration and creativity skills through rubrics: Experience from UTPL collaborative social networks course. *Education Engineering (EDUCON), 2010 IEEE*, 1511-1516. doi:<http://dx.doi.org/10.1109/EDUCON.2010.5492349>

Redecker, C. & Johannessen, Ø. (2013). Changing assessment—Towards a new assessment paradigm using ICT. *European Journal of Education*, 48(1), 79-96. doi:<http://dx.doi.org/10.1111/ejed.12018>

Redecker, C., Punie, Y. & Ferrari, A. (2012). eAssessment for 21st century learning and skills. En *21st Century Learning for 21st Century Skills* (págs. 292-305). Springer Berlin Heidelberg. doi:http://dx.doi.org/10.1007/978-3-642-33263-0_23

Robles, G., Koch, S. & González-Barahona, J. M. (2004). Remote analysis and measurement of libre software systems by means of the CVSAnalY tool. *Proceedings of the 2nd ICSE Workshop on Remote Analysis and Measurement of Software Systems (RAMSS)*, (pp. 51-55). doi:<http://dx.doi.org/10.1049/ic:20040351>

Sadler, D. (2010). Beyond feedback: Developing student capability in complex appraisal. *Assessment & Evaluation in Higher Education*, 35(5), 535-550. doi:<http://dx.doi.org/10.1080/02602930903541015>

Shih, R. C. (2011). Can Web 2.0 technology assist college students in learning English writing? Integrating Facebook and peer assessment with blended learning. *Australasian Journal of Educational Technology*, 27(5), 829-845. doi:<http://dx.doi.org/10.14742/ajet.934>

Vassiliadis, P. (2009). A survey of extract-transform-load technology. *International Journal of Data Warehousing and Mining*, 5(3), 1-27. doi:<http://dx.doi.org/10.4018/jdwm.2009070101>

Walvoord, B. E. & Anderson, V. J. (2009). Effective grading: A tool for learning and assessment in college. *San Francisco: John Wiley & Sons*.

Wood, E. J. (2008). The E-assessment handbook by Geoffrey Crisp. *Biochemistry and Molecular Biology Education*, 36(1), 91-92. doi:<http://dx.doi.org/10.1002/bmb.131>

Zafra, A., Gibaja, E., Luque, M. & Ventura, S. (2011). An Evaluation of the Effectiveness of e-learning system as Support for Traditional Classes. *7th International Conference on Next Generation Web Services Practices (NWeSP)* (pp. 431-435). IEEE. doi:<http://dx.doi.org/10.1109/nwesp.2011.6088218>

Zhao, L. & Elbaum, S. (2003). Quality assurance under the open source development model. *Journal of Systems and Software*, 66(1), 65-75. doi:[http://dx.doi.org/10.1016/S0164-1212\(02\)00064-X](http://dx.doi.org/10.1016/S0164-1212(02)00064-X)