

Una propuesta para la evaluación de proyectos en un curso de Compiladores con una metodología de aprendizaje basada en proyectos

José Miguel Benedí¹ y Emilio Vivancos¹

¹Universitat Politècnica de València

Abstract

In this work, a evaluation system for compilation projects in a course on programming languages and compilers is proposed. In this course we have chosen an active learning methodology aimed at the implementation of a compiler project. Our evaluation system meets the following objectives: to allow a global evaluation of the final project performance; to take into account the individual work of the components of the same team; and to consider continued work on the project. To achieve these objectives, we have proposed: a user simulation model; an individual practical test; and a set of monitoring activities. Finally, we also present a statistical analysis of the results of the last five years. The results of this analysis seem to support the proposed evaluation system.

Keywords: Education, Project-based Learning, Compilers.

Resumen

En este trabajo se propone un sistema integral de evaluación de proyectos de compilación para la asignatura de “Lenguajes de Programación y Procesadores del Lenguaje”. En esta asignatura hemos optado por una metodología activa orientada a la realización de un proyecto. El sistema de evaluación propuesto atiende a los siguientes objetivos: permitir una evaluación global del desempeño final del proyecto; tener en cuenta el trabajo individual de los componentes de un mismo equipo; y considerar el trabajo continuo en el desarrollo del proyecto. Para ello hemos propuesto: un modelo de simulación de usuario; una prueba práctica individual; y un conjunto de actividades de seguimiento. Finalmente, también presentamos un análisis estadístico de los resultados de los últimos cinco años. Los resultados de este análisis parecen avalar el sistema de evaluación propuesto.

Keywords: Educación, Aprendizaje Basado en Proyectos, Compiladores.

Introducción

El curso de Compiladores se imparte actualmente en la asignatura de “*Lenguajes de Programación y Procesadores del Lenguaje*” (LPyPL), del grado en Ingeniería Informática, en la Escuela Técnica Superior de Ingeniería Informática, de la Universitat Politècnica de València (UPV). LPyPL está enmarcada en el módulo de tecnología específica de *Computación* (4º curso, semestre A) y tiene asignados 6 créditos (4,5 teóricos y 1,5 prácticos). El objetivo principal de LPyPL es que el alumno conozca los fundamentos teóricos y prácticos, así como las técnicas y herramientas básicas, para el diseño y construcción de un compilador. Al mismo tiempo se pretende que el alumno sea capaz de aplicar las ideas y técnicas propias del diseño de compiladores en otros campos de la informática.

Recordemos que un compilador acepta como entrada un programa escrito en un cierto *Lenguaje Fuente* (usualmente un lenguaje de programación de alto nivel), y genera un programa escrito en un *Lenguaje Objeto* (usualmente un lenguaje máquina) (AHO y col. 2008). Tradicionalmente, en las asignaturas de compiladores se ha hecho hincapié en aspectos formales de la *Fase de Análisis* (análisis léxico-sintáctico y traducción dirigida por la sintaxis) (WAITE 2006; AHO y col. 2008). Sin embargo, poco a poco se ha tomado conciencia de que centrarse sólo en los aspectos teóricos no ayuda a la comprensión del modo en que debe construirse un compilador útil (AHO 2008; COOPER y TORCZON 2012). Esta observación fue positivamente reforzada por nuestra experiencia en la impartición de las sucesivas asignaturas de compiladores en la UPV (VIVANCOS y col. 1998; BENEDI y col. 2008). A lo largo de estos años, hemos ido modificando gradualmente nuestro punto de vista poniendo el énfasis en la *Fase de Síntesis* (generación y optimización de código) (COOPER y TORCZON 2012). Como resultado de todo ello, el objetivo de la actual asignatura de LPyPL es que los alumnos tengan la base teórica necesaria que les permita construir un compilador completo, que funciona realmente, para un lenguaje de programación de alto nivel, sencillo, pero no trivial.

Para cumplir este objetivo, en LPyPL hemos optado por una estrategia metodológica activa orientada a la realización de un proyecto: *aprendizaje basado en proyectos* (ABP) (RODRIGUEZ-MESA, KOLMOS y GUERRA 2017; BUCK INSTITUTE FOR EDUCATION 2019). Este tipo de aproximación metodológica es muy atractiva ya que los alumnos se enfrentan a situaciones y problemas realistas que deben solucionar en un proceso colectivo de investigación dirigida a la consecución de un objetivo común, en nuestro caso, desarrollar un proyecto de compilación. Además, la labor del profesor ya no es solo la de transmisión de conocimientos, sino que también es la de preparación de actividades que permitan a los alumnos participar en la (re)construcción de los conocimientos y la de dirigir dichas actividades (RODRIGUEZ-MESA, KOLMOS y GUERRA 2017).

En la aplicación de esta estrategia de ABP se han encontrado una serie de ventajas (GALEANA 2006) que constituyen la motivación que justifica esta elección metodológica y que podemos resumir:

- Conseguir que el alumno adquiriera una visión más realista del funcionamiento de un compilador.
- Aumentar la motivación del alumno, ya que se le plantean auténticos retos reales: construcción de un compilador.

- Facilitar la comprensión de algunos conceptos que difícilmente se entenderían solo en las sesiones teóricas.
- Incidir en que esta alternativa es la más cercana al tipo de trabajos que el ingeniero en informática se va a encontrar en sus labores profesionales.
- Desarrollar capacidades transversales como el trabajo en equipo, las relaciones interpersonales, la comunicación, toma de decisiones y manejo del tiempo.

Sin embargo, la puesta en marcha de una estrategia de ABP se enfrenta, sin duda, a grandes inconvenientes. A lo largo del tiempo, hemos detectado 2 grandes fuentes de problemas: sobrecarga de trabajo, tanto para los alumnos como para los profesores, y evaluación justa del proyecto, tanto individual como grupal.

Respecto al primer problema, la *sobrecarga de trabajo para los alumnos* es uno de los aspectos críticos asociados con el ABP y obliga a medir cuidadosamente el esfuerzo exigido a los alumnos y a estructurar adecuadamente el proyecto. Por otro lado, la *sobrecarga de trabajo para los profesores* está relacionado con el enorme esfuerzo y tiempo de dedicación que el profesorado debe invertir en el diseño y la preparación del proyecto de compilación para cada curso académico. La preparación, por parte del profesorado, de un nuevo proyecto de compilación lleva asociadas varias e importantes tareas: definir las especificaciones formales (léxicas, sintácticas y semánticas) del lenguaje fuente del proyecto; decidir, definir y desarrollar el lenguaje objeto, así como de su máquina virtual; diseñar, implementar, probar y documentar todo el material de ayuda (librerías); y por último, elaborar toda la documentación necesaria.

Para mitigar la sobrecarga de trabajo de los alumnos y de los profesores propusimos un entorno de desarrollo, de libre disposición y portable, para facilitar el diseño y la elaboración de nuevos proyectos de compilación (BENEDÍ y VIVANCOS 2016). Este entorno ha sido utilizado con éxito en los últimos cursos de LPyPL.

El segundo problema detectado está relacionado con la *evaluación del proyecto*. En la literatura existe un consenso general en la importancia de la evaluación del proyecto en una estrategia de ABP y hay una gran cantidad de trabajos previos que abordan este tema (PÉREZ-GONZÁLEZ y col. 2008; RODRIGUEZ-MESA, KOLMOS y GUERRA 2017; BUCK INSTITUTE FOR EDUCATION 2019). Algunas de estas propuestas las hemos venido aplicando en LPyPL (VIVANCOS y col. 1998; BENEDI y col. 2008) con resultados no demasiado satisfactorios. En este trabajo vamos a presentar un nuevo sistema integral de evaluación de proyectos (de compilación) atendiendo a tres consideraciones:

1. *Evaluación objetiva del proyecto*. En el caso de LPyPL, el proyecto es un compilador real, por tanto, consideramos que su evaluación también debe ser lo más realista posible y sobre todo atendiendo al desempeño del mismo.
2. *Evaluación discriminativa individual del alumno*. Dado que el proyecto se realiza en grupos de 3 a 4 alumnos, el problema de la evaluación individual de la aportación de cada alumno al desarrollo final del proyecto cobra, a nuestro entender, una gran importancia en la motivación final del alumno.
3. *Evaluación continua del trabajo del grupo*. El proyecto está dividido en fases o etapas que los componentes del grupo deben cubrir hasta alcanzar el objetivo final. Por tanto, además de evaluar el resultado final, consideramos que se debe evaluar del proceso continuo de desarrollo del proyecto.

En la [Sección 2](#), mostraremos brevemente el entorno de desarrollo de proyectos de compilación. En la [Sección 3](#), enunciaremos los objetivos que pretendemos alcanzar con la propuesta. Teniendo en cuenta las consideraciones expuestas sobre la evaluación, en la [Sección 4](#), presentaremos una nueva propuesta de evaluación de proyectos tanto individual como grupal. Finalmente, en la [Sección 5](#), valoraremos el resultado de la aplicación de esta propuesta de evaluación en la asignatura de LPyPL de la UPV.

Entorno de desarrollo de proyectos de compilación

El objetivo de este entorno de desarrollo de proyectos de compiladores (EDPC) es doble: por un lado, reducir la carga de trabajo a los alumnos, proporcionándoles un material de ayuda accesible y completo; y por otro, simplificar del proceso de creación de nuevos proyectos y favorecer la elaboración de proyectos de distinta naturaleza y amplitud. EDPC está pensado para que sea fácil de modificar y ampliar, y permita diseñar proyectos modulares, incrementales y portables (BENEDÍ y VIVANCOS 2016). El EDPC completo se compone de una serie de módulos que presentamos a continuación:

1. **Lenguaje fuente.** La elección del lenguaje fuente incorpora todos los problemas de diseño e implementación de un lenguaje de programación y debe satisfacer dos consideraciones previas: 1) que el proyecto esté bien definido, es decir, que se conozcan las especificaciones léxicas, sintácticas y semánticas del lenguaje fuente; y 2) que el proyecto sea tratable, es decir, que sea suficientemente reducido como para que los alumnos puedan implementar su compilador en un semestre (BENEDÍ y VIVANCOS 2016).
2. **Lenguaje objeto y máquina virtual.** El EDPC nace con la voluntad de ser independiente de la plataforma, portable y fácil de instalar en cualquier máquina UNIX con herramientas estándar de software GNU. Para alcanzar este objetivo, en el EDPC, nos hemos decantado por un código intermedio 3-direcciones (lenguaje objeto), independiente de la máquina (AHO y col. 2008) y similar al que se emplea en las clases de teoría, ya que es simple e intuitivo y se adapta perfectamente a las necesidades del proyecto de compilación para la asignatura de LPyPL.

Para poder evaluar en condiciones reales el comportamiento del compilador se ha diseñado y construido una máquina virtual que es capaz de ejecutar el código intermedio generado por el compilador del proyecto (BENEDÍ y VIVANCOS 2016).

3. **Material de apoyo.** En la construcción de un compilador es necesario codificar muchas estructuras de datos (p.ej. la tabla de símbolos) y generar código repetitivo de bajo nivel (p.ej. las plantillas para la generación de código intermedio). EDPC proporciona todo este código adicional, debidamente documentado, en forma de dos librerías (BENEDÍ y VIVANCOS 2016):
 - **libtds.** Librería con las operaciones para la correcta manipulación de la tabla de símbolos.
 - **libgci.** Librería con las operaciones para la gestión de memoria y la generación de código intermedio.

Facilitar este material de apoyo tiene otras ventajas adicionales: proporciona un nivel moderado de abstracción, elimina en gran medida los posibles errores y permite a los estudiantes centrarse en los aspectos más importantes del proyecto.

4. **Compilador de referencia: caso de estudio.** El EDPC también aporta un ejemplo o caso de estudio sencillo y derivado directamente de los ejercicios parciales realizados en las clases de teoría. Este compilador de referencia sirve como ejemplo sencillo de uso del EDPC para resolver un problema conocido y como ayuda para ilustrar la utilización del material de apoyo y las librerías. Las partes en las que se descompone el compilador pueden compilarse y analizarse por separado, lo que apoya el proceso incremental de la construcción del compilador.
5. **Documentación.** La documentación que se aporta a los alumnos para la adecuada elaboración del proyecto se puede resumir:
 - Especificación formal (léxica, sintáctica y semántica) para lenguaje fuente del curso actual.
 - Descripción completa del material de apoyo, así como de las restricciones de su uso.
 - Guía para la elaboración del proyecto de compiladores.
 - Documentación adicional de todas las herramientas software utilizadas en el proyecto: Generador Automático de Analizadores Léxicos (FLEX)¹ y Generador Automático de Analizadores Sintáctico-Semánticos (BISON).²
 - Documentación y código fuente completo del compilador de referencia para el caso de estudio.
6. **Programas de prueba para la autoevaluación.** Estos programas de prueba permiten a los alumnos la correcta autoevaluación de cada una de las etapas en las que se divide su compilador: análisis léxico-sintáctico, análisis semántico y generación de código (intermedio). En la [Sección 4](#), incidiremos más en este aspecto de la autoevaluación.

Objetivos de la propuesta

La evaluación es una pieza fundamental para el buen funcionamiento del método de ABP y, en general, para cualquier trabajo realizado en equipo. Debido a ello, la evaluación de un proyecto ha sido tratada con profusión en la literatura (PÉREZ-GONZÁLEZ y col. 2008; RODRIGUEZ-MESA, KOLMOS y GUERRA 2017; BUCK INSTITUTE FOR EDUCATION 2019), constatando la enorme dificultad de dicha tarea. A nuestro juicio, dos son los grandes retos que comporta la evaluación de un proyecto: cómo realizar una evaluación realista del proyecto y cómo evaluar individualmente a los diferentes componentes de un mismo equipo. No existe un acuerdo a la hora de diseñar un plan de evaluación que aborde estos retos. Algunas alternativas comunes son: el uso de portafolios individuales, la autoevaluación, la evaluación entre pares dentro del

¹FLEX: <https://github.com/westes/flex>

²BISON: <https://www.gnu.org/software/bison/>

grupo, la evaluación entre grupos, etc. Cada una de ellas presenta ventajas e inconvenientes pero adolecen de alguna fragilidad debido a una cierta subjetividad, tanto en la valoración del desempeño del proyecto, como en la evaluación individual de la participación de los alumnos en el proyecto. Ambos problemas pueden introducir una cierta falta de realismo y la consiguiente pérdida de motivación en los alumnos.

Con todo ello, en el contexto de las metodologías activas orientadas a la realización de un proyecto, nuestra propuesta de un nuevo sistema de evaluación de proyectos (compiladores) para la asignatura de LPyPL considera los siguientes objetivos principales:

- Debe permitir una evaluación global del desempeño final del proyecto.
- Debe tener en cuenta el trabajo individual de los componentes de un mismo equipo.
- Debe considerar el trabajo continuo en el desarrollo del proyecto.

A estos objetivos principales se le añaden también otros objetivos adicionales:

- Debe responder a los objetivos formativos planteados, tanto los específicos de cada asignatura como los transversales.
- Debe ser claro e informativo para los estudiantes, de modo que éstos sepan los aspectos concretos por los que se les va a evaluar.
- Debe fomentar la capacidad de evaluación y autoevaluación de los estudiantes.

Desarrollo de la innovación

Un sistema de evaluación que esté bien diseñado emplea diversas evidencias para determinar si los estudiantes han cumplido con los objetivos del proyecto. Para el proyecto de compiladores de la asignatura de LPyPL las evidencias de evaluación consideradas son las siguientes:

Evaluación del desempeño final del proyecto

Tal y como se apuntó en las secciones anteriores, uno de los objetivos principales de una metodología ABP es que el proyecto debe ser lo más realista posible para que pueda ilustrar correctamente el tipo de trabajos relacionados con esta asignatura. Además se espera que esto incremente la motivación de los alumnos para realizar el proyecto. Por esas mismas razones, la evaluación del resultado final del proyecto también debería ser realista; es decir, debería evaluarse por medio de un usuario final que verifique objetivamente si el desempeño final del proyecto cumple con las especificaciones de diseño.

Dada la naturaleza de la asignatura de LPyPL, y dado que el proyecto es un compilador, el usuario final debería verificar su desempeño final por medio de la compilación y ejecución de un subconjunto de programas de prueba. Considerando todo ello, y teniendo en cuenta que los alumnos son usuarios habituales de compiladores para sus lenguajes de programación favoritos, en este trabajo proponemos un **modelo de simulación de usuario** formado por dos elementos:

- Un conjunto de programas de prueba, seleccionados por los profesores, para que se pueda testear cada una de las partes que integran el compilador.
- Una máquina virtual, incluida en el EDPC, para verificar si el compilador cumple las especificaciones de diseño, para permitir ejecutar el código generado y para demostrar su corrección.

Además, este modelo favorece la autoevaluación de los diferentes grupos de trabajo, ya que ellos son los usuarios finales y tienen las herramientas para analizar el desempeño de las diferentes partes de su proyecto.

Evaluación individual del alumno

En la puesta en marcha del ABP para LPyPL, el proyecto de compilación se realiza en grupos. Esto plantea una gran dificultad a la hora de realizar la evaluación (PÉREZ-GONZÁLEZ y col. 2008; RODRIGUEZ-MESA, KOLMOS y GUERRA 2017). En nuestro caso, creemos que es necesaria una evaluación individualizada de las aportaciones de cada alumno del grupo en el desarrollo del proyecto, y que ésta debe ser justa y objetiva.

En este trabajo proponemos una **prueba práctica individual**, realizada en el laboratorio, y donde cada alumno debe modificar el código de su compilador para incorporar una muy pequeña modificación en el lenguaje fuente.

Esta prueba individual valora efectivamente el grado de comprensión y de implicación del alumno en el proyecto. Además, dado que esta prueba es conocida desde el comienzo del curso, motiva a todos los alumnos a participar activamente en la elaboración de su compilador. Sin embargo, existe la posibilidad de que el miedo a esta prueba pueda retraer a los alumnos menos implicados y provocar un abandono prematuro del proyecto. Para evitar este problema, se han tomado una serie de medidas correctoras: tutorización, tanto grupal como individual, en las clases de laboratorio y propuesta de posibles pruebas de ejemplo para que los alumnos puedan evaluar con anterioridad la dificultad de las mismas. En [Sección 5](#) mostraremos que estas medidas correctoras han funcionado satisfactoriamente.

Evaluación continua

En una estrategia de ABP, además de la evaluación global del proyecto, también se debe valorar el trabajo continuo de los alumnos en la búsqueda de información y resolución de los sucesivos problemas que se les van presentando en la resolución de su proyecto (GALEANA 2006; RODRIGUEZ-MESA, KOLMOS y GUERRA 2017; BUCK INSTITUTE FOR EDUCATION 2019).

En LPyPL, para facilitar la tarea de implementación y verificación del proyecto de compilación, éste se divide en tres etapas: analizador léxico-sintáctico, analizador semántico, y generador de código intermedio. Para evaluar el trabajo continuo en la construcción del compilador, en esta propuesta definimos un conjunto de **actividades de seguimiento** en el laboratorio:

- *Evaluación continua*, mediante los correspondientes entregables asociados con cada una de las partes del proyecto. Estos entregables solo constan del código de cada una de las partes del compilador.
- *Autoevaluación continua*, mediante la superación de los programas de prueba, proporcionados por los profesores, de cada una de las partes del compilador
- *Tutorización*, que permite conocer la evolución del compilador tanto del grupo en su conjunto como de cada uno de los integrantes del mismo.

Resultados y valoración de la experiencia

Esta experiencia la hemos llevado a cabo durante los 5 últimos cursos académicos (2014-15 al 2018-19) y ha afectado tanto al trabajo de los profesores como al desempeño de los alumnos. Desde el punto de vista del profesorado, la valoración de la experiencia ha sido muy positiva ya que hemos podido constatar la facilidad y la considerable reducción del trabajo necesario para la elaboración anual de cada nuevo proyecto de compilación para LPyPL. Desde el punto de vista de los alumnos, el EDPC también ha supuesto una considerable ayuda como lo demuestran algunos indicadores que vamos a presentar a continuación:

Porcentaje de proyectos entregados y evaluados positivamente

En la primera fila de la [Tabla 1](#) (proyectos entregados/alumnos matriculados) se puede observar un elevado porcentaje de alumnos que han completado su proyecto (89,4%, en promedio) desde la implantación del EDPC. Los alumnos que no lo consiguen, típicamente son alumnos que, por diversos motivos, se dejan la asignatura y no se presentan a ningún acto de evaluación.

Esta medida nos indica, de una manera indirecta, el esfuerzo requerido a los alumnos para la elaboración de su proyecto. Si fuera excesivo, los alumnos tienden a abandonarlo en las primeras fases del proyecto. Mientras que si cuentan con la ayuda necesaria, hemos observado que los alumnos se implican más y se mantienen activos hasta la finalización del proyecto. Lo que nos permite concluir que el trabajo exigible a los alumnos para la elaboración de su proyecto es perfectamente asumible.

Tabla 1: Porcentaje de proyectos entregados y evaluados positivamente

	2014-15	2015-16	2016-17	2017-18	2018-19
<i>Entregados / Matriculados</i>	91 %	85 %	88 %	87 %	96 %
<i>Aprobados / Entregados</i>	100 %	95 %	90 %	91 %	98 %

En la segunda fila de la [Tabla 1](#) (proyectos aprobados/proyectos entregados) se muestra el porcentaje de alumnos que han sido evaluados positivamente en su proyecto respecto a los que lo han completado. Como puede observarse, este porcentaje es muy alto (94,8%). Estos excelentes resultados nos muestran que, no solo la dificultad del proyecto está razonablemente ajustada, sino que el proceso de evaluación también considera el trabajo continuo en el desarrollo del proyecto, ya que la inmensa mayoría de los alumnos activos en el proyecto completan los actos de evaluación planteados.

Porcentaje de alumnos presentados al examen individual de prácticas

A continuación hemos realizado un análisis estadístico de los resultados de los últimos cinco años respecto a los alumnos presentados al examen individual de prácticas. En la [Figura 1](#), lo primero que se observa es el alto porcentaje de alumnos presentados (91 % en promedio). Esta medida, de alguna manera, da cuenta del grado de comprensión del proyecto que tiene el alumno. Solo los alumnos que han participado activamente en el desarrollo del proyecto de su grupo suelen presentarse a este examen individual del proyecto.

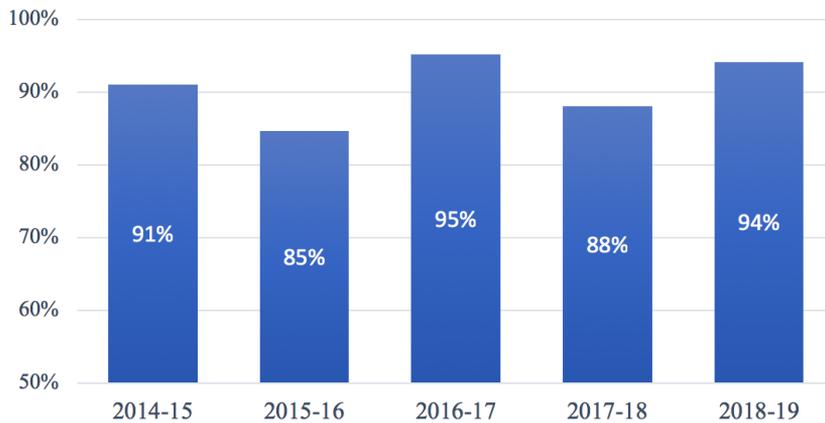


Fig. 1: Porcentaje de alumnos presentados al examen individual de prácticas.

Importancia del proyecto en la comprensión de la asignatura

En este punto vamos a analizar la correlación que hay entre los alumnos que hacen el proyecto correctamente y las notas que sacan de teoría. La idea es comprobar la importancia del proyecto en la comprensión de la asignatura y no solo en la parte práctica. Al comparar las calificaciones del examen individual de prácticas con los del examen de teoría, se obtiene un coeficiente de correlación $\rho_{x,y} = 0,41$, lo que indica la clara relación que hay entre los resultados del examen individual de prácticas y el de teoría. Podemos estudiar con más detalle esta relación, analizando los resultados en el examen de teoría del conjunto de alumnos que han aprobado el examen de prácticas. En la [Figura 2](#) puede observarse que, en todos los años estudiados, un alto porcentaje de los alumnos que aprobaron el examen individual de prácticas, también aprobaron el examen de teoría.

A modo de resumen se muestran en la [Figura 3](#) y [Figura 4](#) los resultados que obtuvieron, en el examen de teoría, el total de alumnos de los últimos 5 cursos agrupados en dos conjuntos: los resultados de los alumnos que aprobaron el examen individual de prácticas, [Figura 3](#), ($\approx 75\%$ del total); y los de los que suspendieron el examen individual de prácticas, [Figura 4](#), ($\approx 25\%$ del total). Los alumnos que aprobaron el examen individual de prácticas, mayoritariamente (73%) aprobaron también el examen de teoría, mientras que los alumnos que suspendieron el examen individual de prácticas, mayoritariamente (72%) también suspendieron el examen de teoría.

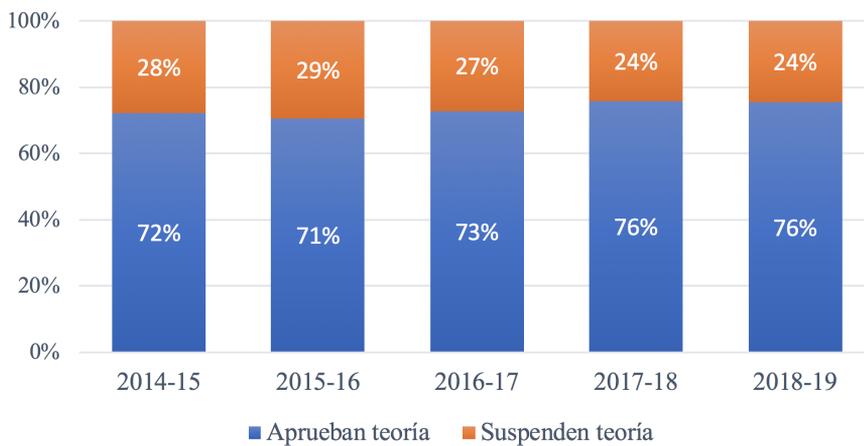


Fig. 2: Resultados del examen de teoría del grupo de alumnos que aprobaron el examen de prácticas.

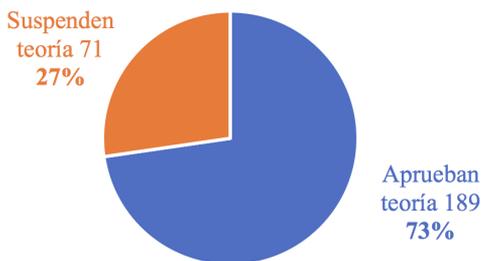


Fig. 3: Resultados de teoría de los alumnos que aprobaron prácticas.

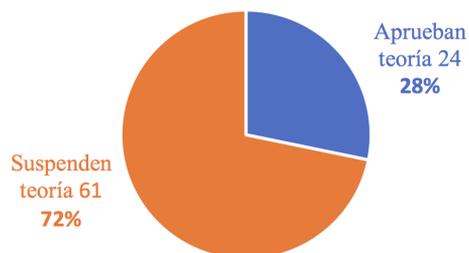


Fig. 4: Resultados de teoría de los alumnos que suspendieron prácticas.

Por último, indicar que del grupo de alumnos que aprobaron el examen individual de prácticas un 97,2% aprobaron también la asignatura. Mientras que los alumnos que suspendieron el examen individual de prácticas, un 72,5% suspendieron también la asignatura.

Esta correlación entre los resultados del examen individual de prácticas y del examen de teoría se observa también en las calificaciones medias de la nota de teoría y la nota final de la asignatura, en función de si se ha aprobado o no el examen individual de prácticas (Figura 5). Como puede observarse, el grupo de alumnos que aprobaron el examen individual de prácticas obtuvo en el examen de teoría una calificación media superior en 1,9 puntos a la del grupo de alumnos que suspendió el examen de prácticas. En el caso de la nota final de la asignatura, la diferencia es mucho mayor (3,4 puntos). Lo que nos indica la importancia del proyecto en el resultado final de asignatura.

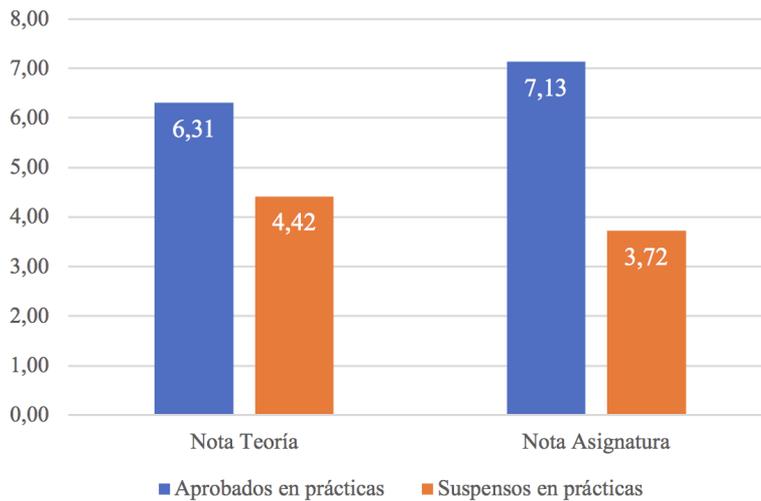


Fig. 5: Nota media de teoría y nota media final de la asignatura, de los alumnos aprobados y suspendidos en el examen individual de prácticas de los últimos 5 cursos (2014-15 al 2018-19).

Conclusiones

En este trabajo hemos presentado un sistema integral de evaluación de proyectos de compilación para la asignatura de LPyPL de la UPV. Este sistema de evaluación atiende a los siguientes objetivos: permitir una evaluación global del desempeño final del proyecto, tener en cuenta el trabajo individual de los componentes de un mismo equipo, y considerar el trabajo continuo en el desarrollo del proyecto. Para lo cual hemos propuesto: un modelo de simulación de usuario, una prueba práctica individual, y un conjunto de actividades de seguimiento.

Del análisis de los resultados en los 5 últimos años se puede destacar: 1) la dificultad y el esfuerzo exigido a los alumnos es adecuado, ya que un porcentaje muy elevado de alumnos completan el proyecto; 2) el grado de comprensión del proyecto es alto, ya que el porcentaje de alumnos que se presentan al examen individual de prácticas ronda el 91% en promedio; y 3) la importancia del proyecto en la comprensión de la asignatura es muy destacada, como lo atestiguan las notas medias en teoría de alumnos aprobados en el examen individual de prácticas.

No obstante, la puesta en marcha de este sistema de evaluación de proyectos requiere de una importante infraestructura y una complicada logística para la preparación del examen individual de prácticas. Esto es, se necesita un computador (fuera de línea) por alumno y además se precisa de un software específico que gestione la correcta entrega del resultado de dicho examen individual. En nuestro caso todo esto es posible porque el Departamento de Sistemas Informáticos y Computación dispone de la cantidad suficiente de laboratorios y computadores que se necesitan para hacer el examen individual de prácticas y unos técnicos de laboratorio perfectamente capacitados para gestionarlos. Sin embargo, somos conscientes que estos requerimientos constituyen una posible limitación a la hora de poder exportar a otras asignaturas esta propuesta de evaluación de proyectos en una metodología activa de ABP.

Referencias bibliográficas

- AHO, Alfred (2008). “Teaching the compilers course”. En: *ACM SIGCSE Bulletin* 40.4, págs. 6-8.
- AHO, Alfred y col. (2008). *Compilers: Principles, Techniques, and Tools (2nd Edition)*. Addison Wesley.
- BENEDÍ, José Miguel y Emilio VIVANCOS (2016). “Un entorno para el desarrollo de proyectos en la enseñanza activa de un curso de Compiladores”. En: *II Congreso Nacional de Innovación Educativa y Docencia en Red*. Universitat Politècnica de València, págs. 1-11.
- BENEDI, José Miguel y col. (2008). *Procesadores de Lenguajes: una introducción a la fase de análisis*. Universitat Politècnica de València.
- BUCK INSTITUTE FOR EDUCATION (2019). *Project Based Learning*. Boise State University. URL: <http://pbl-online.org/>.
- COOPER, Keith y Linda TORCZON (2012). *Engineering a Compiler*. Morgan Kaufman.
- GALEANA, Lourdes (2006). “Aprendizaje basado en proyectos”. En: *Investigación y Desarrollo Pedagógico: Revista digital*.
- PÉREZ-GONZÁLEZ, Antonio y col. (2008). “Un sistema para la evaluación del aprendizaje basado en proyectos”. En: *XVI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas*. Cádiz.
- RODRIGUEZ-MESA, Fernando, Anette KOLMOS y Aida GUERRA, eds. (2017). *Aprendizaje basado en problemas en ingeniería: Teoría y práctica*. Aalborg University Press.
- VIVANCOS, Emilio y col. (1998). “Metodología docente orientada a proyectos aplicada a las prácticas de compiladores”. En: *IV Jornades sobre l'ensenyament universitari de la informàtica*. Sant Julià de Lòria, Principat d'Andorra, págs. 480-484.
- WAITE, William M. (2006). “The Compiler Course in Today's Curriculum: Three Strategies”. En: *SIGCSE Bull.* 38.1, págs. 87-91.