



Innovación docente para el desarrollo de la competencia transversal “Conocimiento de problemas contemporáneos” en el marco de una asignatura de Calidad de Software

Silvia Abrahão^a, Emilio Insfran^a

^aUniversitat Politècnica de València ({sabrahao,einsfran }@dsic.upv.es)

Abstract

This paper presents a teaching innovation based on the implementation of different training activities for the development of the competency "CT-10 Knowledge of Contemporary Problems" of the Transversal Competences Project of the Universitat Politècnica de València (UPV). The innovation is based on the use of different training activities conducted in the context of a course on Software Quality of the Bachelor's Degree in Informatics Engineering at the UPV. In particular, Projects, Report Writing and Seminar have been carried out to present the results of practical activities, close to the work that will be conducted by a software quality engineer in a business environment. We present the results of the competence acquisition by the students. A survey has also been carried out in order to assess the students' opinion about the teaching activities. Finally, we describe our lessons learned and possible improvements to the training activities based on the experience gained.

Keywords: *Transversal competences, Knowledge of contemporary problems, Computer Science, Learning activities*

Resumen

En este artículo se presenta una innovación docente basada en la puesta enmarcha de diferentes actividades formativas para poder desarrollar la competencia “CT-10 Conocimiento de problemas contemporáneos” del Proyecto de Competencias Transversales de la Universitat Politècnica de València (UPV). La innovación se basa en la utilización de diferentes actividades formativas dentro de la asignatura Calidad de Software del Grado de Ingeniería Informática de la UPV. En particular, se ha realizado proyectos, redacción de informes y exposición oral para presentar los resultados de actividades de carácter práctico cercanas al trabajo que realizaría un ingeniero de calidad de software en un entorno profesional. Se presentan los resultados de la adquisición de la competencia por parte de los alumnos y de una encuesta realizada para conocer su opinión sobre las actividades docentes. Finalmente, se describen las lecciones aprendidas y posibles mejoras al programa de actividades formativas en base a la obtenida.

Palabras clave: *Competencias transversales, Conocimiento de problemas contemporáneos, Ingeniería Informática, Actividades formativas.*

1. Introducción

El Plan Estratégico 2015-2020 y el Proyecto Institucional de Competencias Transversales de la Universitat Politècnica de València (UPV) tiene como objetivo ofrecer una formación integral a los estudiantes que abarque no solo el aprendizaje de contenidos específicos de las materias impartidas, sino también la adquisición de destrezas, habilidades y valores transversales. Se pretende que los alumnos desarrollen tanto competencias específicas propias de su título como competencias transversales que utilizarán tanto en su vida profesional como en su vida personal.

En concreto, la UPV definió trece Competencias Transversales (Tabla 1), con el fin de agrupar los distintos perfiles competenciales y poder abordar el problema de trabajo y evaluación de las distintas competencias transversales en todas las titulaciones que imparte.

Tabla 1. Competencias Transversales UPV

Código	Tipo
CT-01	Comprensión e integración
CT-02	Aplicación y pensamiento práctico
CT-03	Análisis y resolución de problemas
CT-04	Innovación, creatividad y emprendimiento
CT-05	Diseño y proyecto
CT-06	Trabajo en equipo y liderazgo
CT-07	Responsabilidad ética, medioambiental y profesional
CT-08	Comunicación efectiva
CT-09	Pensamiento crítico
CT-10	Conocimiento de problemas contemporáneos
CT-11	Aprendizaje permanente
CT-12	Planificación y gestión del tiempo
CT-13	Instrumental específica

Estas competencias transversales tienen asociado una serie de resultados de aprendizaje, agrupados en dos niveles (grado y máster). Sin embargo, es necesario evaluar y obtener evidencias de estas competencias en las distintas titulaciones (Oltra y Gil 2017). En este artículo, se presenta una propuesta de actividades formativas que, de forma integrada en la asignatura, van a apoyar el desarrollo de la CT-10, Conocimiento de problemas contemporáneos en la asignatura de Calidad de Software del Grado de Ingeniería Informática, en la E.T.S de Ingeniería Informática de la UPV.

Esta competencia transversal hace referencia a la necesidad de que los alumnos comprendan las cuestiones y valores contemporáneos, así como los mecanismos de

expansión y difusión del conocimiento. Se trata de que desarrollen la capacidad de estar al día en eventos actuales en su campo de especialización como en otros campos de conocimiento.

La formación en competencias implica tres fases. La primera primera fase se centra en los profesores y en el diseño de los instrumentos y materiales para el trabajo de la competencia. La segunda fase involucra también a los alumnos e incluye el uso de estos materiales en la asignatura. Finalmente, la tercera fase, consiste en la evaluación de las actividades formativas llevadas a cabo con los alumnos y el uso de los resultados para mejorar los instrumentos y materiales.

En este artículo, las competencias específicas que se trabajan están vinculadas al desarrollo de un modelo de calidad alineado con las normas ISO/IEC 25010 y ISO/IEC 25040 para la evaluación y control de calidad de sistemas software contemporáneos. El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado. En la asignatura, los alumnos tienen que escoger un dominio específico sobre el que trabajar y desarrollar el modelo para este dominio usando como base las normas mencionadas anteriormente.

Este artículo se estructura como sigue. La sección 2 describe los objetivos del artículo y el contexto en el que se desarrolla. La sección 3 describe el desarrollo de la innovación para incorporar la competencia CT10 en la asignatura de Calidad de Software. La sección 4 describe los resultados obtenidos. Finalmente, la sección 5 presenta las conclusiones y trabajos futuros.

2. Objetivos

Los objetivos del artículo son:

1. Analizar la competencia transversal de “conocimiento de problemas contemporáneos”.
2. Establecer indicadores para medir su desarrollo y aplicación.
3. Proponer mecanismos para evaluar la competencia.
4. Mostrar su aplicación a una asignatura de grado de Ingeniería Informática de la UPV.
5. Reflexionar sobre posibles mejoras sobre como abordar la competencia en este contexto.

2.1 Contexto: asignatura Calidad de Software

Se trata de una asignatura opcional que se imparte en el tercer curso del Grado en Ingeniería Informática, dentro de la Rama de Ingeniería del Software. Se ha establecido, dentro del plan de estudios de la titulación, que esta asignatura sea Punto de Control de la competencia CT10. Ser punto de control implica plantear actividades para, en el desarrollo de los contenidos, trabajar la competencia transversal y evaluarla, recogiendo evidencias de los logros alcanzados (UPV 2015a).

La asignatura de Calidad de Software tiene 4,5 créditos repartidos en 3 créditos de teoría y 1,5 créditos de práctica. Se imparte docencia a dos grupos. En este artículo, se describe la experiencia piloto realizada en el Curso 2016-17. Este año el número de matriculados fue de 76 alumnos.

El objetivo de esta asignatura es que el alumno sea capaz de entender y aplicar los principios y técnicas para la gestión, control y aseguramiento de la calidad de productos y procesos software; entender cómo se desarrollan y utilizan los modelos de calidad y las métricas de software; ejecutar inspecciones de productos software; entender los principios y técnicas de estimación de proyectos de software.

Para cumplir con estos objetivos, la asignatura proporciona un conjunto de métodos, técnicas y mecanismos de estimación y aseguramiento de calidad que los alumnos deben valorar e integrar en el desarrollo de sistemas informáticos para así garantizar que los productos software sean entregados en los plazos previstos y con la calidad esperada.

La calidad de un sistema software se puede definir como “el grado con que un sistema de información, componente o proceso cumple con los requisitos especificados y las necesidades o expectativas de los clientes o usuarios”. Según el curriculum SE 2004 de ACM y IEEE, la calidad del software es un concepto pervasivo que afecta (y es afectada) por todos los aspectos del desarrollo, soporte, revisión y mantenimiento de un sistema software. Abarca la calidad de los productos desarrollados y/o modificados tanto de los productos intermedios (modelos conceptuales, esquemas de bases de datos etc.) como de los productos entregables y también la calidad de los procesos utilizados para desarrollar y/o mantener dichos productos.

Por lo tanto, en esta asignatura, se trabajan las siguientes **competencias específicas**:

- **ES1:** Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software.
- **ES4:** Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

Además, se trabajan también las siguientes **competencias generales**:

- **CB5:** Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
- **G04:** Razonar de manera abstracta, analítica y crítica, sabiendo elaborar y defender argumentos en su área de estudio y campo profesional.
- **G05:** Aprender de manera autónoma nuevos conocimientos y técnicas adecuados para la concepción, el desarrollo, la evaluación o la explotación de sistemas informáticos.
- **G12:** Capacidad de integrarse y trabajar eficientemente en equipos unidisciplinarios así como de colaborar en un entorno multidisciplinar.

3. Desarrollo de la Innovación

Durante el curso 2016-2017 se implantaron algunas actividades formativas con el propósito de desarrollar la competencia transversal de “conocimientos de problemas contemporáneos”. Estas actividades se han llevadas a cabo de Febrero a Mayo de 2017.

El objetivo fue familiarizar a los alumnos con la actualidad, estableciendo vínculos entre el material objeto de estudio (calidad del software) y el entorno profesional. Dichas actividades están basadas en metodologías activas para la formación de competencias que fomentan la participación del alumno y generan un aprendizaje más profundo, significativo y duradero (Fernández March, 2006).

El primer paso del desarrollo de la innovación fue *identificar e interpretar los problemas contemporáneos en su campo de especialización, así como en otros campos de conocimiento*. Esto implica que los ingenieros de software deben entender no solo los problemas relacionados con la práctica de su profesión sino también cómo estas prácticas pueden influir en la sociedad de manera general. Se trata de que los alumnos estén al día en cuestiones y valores sociales, legales o medio ambientales que afectan a su area de conocimiento. Esto trasladado a la ingeniería del software podría definirse como la habilidad de que un ingeniero de software sea capaz de:

“Desarrollar, mantener y evaluar sistemas software que satisfagan los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad”.

Estas normas pueden involucrar el hecho de que el software cumpla con aspectos éticos, legales (ley de protección de datos), sociales (que sea accesible por usuarios con discapacidad) o medio-ambientales (que sea respetuoso con el medio ambiente). De hecho, el *World Business Council for Sustainable Development's Vision 2050* (WBCSD, 2019) señala que las TIC (Tecnologías de Información y Comunicaciones) pueden contribuir en gran medida a la respuesta contra el cambio climático a través de soluciones para el ahorro energético y de emisiones en diferentes sistemas como pueden ser el del transporte, las edificaciones, etc.

El desarrollo de software de calidad está altamente ligado a la sostenibilidad del mismo. Un software de calidad implica a un producto usado eficientemente y con un importante ahorro de costes de mantenimiento. Por ejemplo, una característica relevante es la *disponibilidad e integridad* de información en el momento y lugar apropiados y la posibilidad de entender y tener un control efectivo sobre los aspectos apropiados de *seguridad de la información y acceso*. Otra característica de calidad esencial es la *usabilidad del sistema software* o, en otras palabras, la experiencia generada al usuario con el uso de la aplicación. Esta característica de calidad es crítica en el contexto actual en el que el software se ha convertido en un instrumento de uso masivo para usuarios de todo tipo.

2.2 Actividades formativas

En la asignatura de Calidad de Software nos centraremos en *“evaluar sistemas software que satisfagan los requisitos del usuario y cumplan normas de calidad”* ya que los aspectos

de desarrollo y mantenimiento se tratan en otras asignaturas de la rama de Ingeniería del Software. La actividad (Proyecto) para trabajar la CT10 consiste en que los alumnos *definan un Modelo de Calidad para evaluar un tipo de producto software actual* (servicios cloud, aplicaciones web bancarias, videojuegos, aplicaciones gubernamentales, redes sociales, sistemas de transporte, etc.). La Tabla 2 describe la actividad desarrollada.

El proyecto se desarrolla en grupos de 4 personas a lo largo del cuatrimestre. Los propios alumnos se encargan de formar los grupos en función de su afinidad con sus compañeros y su interés por un tema específico. La actividad se ha llevado a cabo en cuatro sesiones de prácticas de laboratorio de 2h cada una y 4 horas de trabajo autónomo, y ha consistido en:

1. Una *fase teórica*, en la cuál los alumnos investigan sobre un tema contemporáneo (las particularidades de un producto software actual y sus implicaciones para los usuarios que los utilizan y su entorno) y proponen un modelo para evaluar la calidad de este producto mediante la descomposición jerárquica de la calidad del producto en características, subcaracterísticas y atributos de calidad. Para esta parte del trabajo los alumnos deben extender el modelo de calidad de software genérico propuesto en la norma ISO/IEC 25010 (2011).
2. Una *fase práctica*, en la cuál los alumnos definen métricas para medir los atributos de calidad y lo aplican a un caso específico. Para esta parte del trabajo, los alumnos deben utilizar el modelo de referencia general para la evaluación del software propuesto en la norma ISO/IEC 25040 (2011).

En particular, la actividad involucra una serie de otras actividades específicas:

1. El profesor propone una lista de posibles temas a trabajar, pero no obstante, se admiten propuestas de nuevos temas por parte de los alumnos.
2. Formación de grupos
3. Los alumnos buscan material acerca del tema y lo expone al profesor con una reflexión personal al respecto.
4. Cada grupo desarrolla su modelo de calidad en varias sesiones prácticas. Las sesiones presenciales son complementadas por trabajo en equipo no presencial. Además, los alumnos entregan un informe por grupo de trabajo al final del curso con el desarrollo del modelo, una aplicación del mismo a un caso práctico y sus conclusiones y reflexiones. Para guiar a los alumnos, el profesor introduce distintos indicadores (ver Sección 2.3) que serán utilizados para evaluar el trabajo.
5. Los alumnos exponen el trabajo oralmente en el aula al profesor y a sus compañeros (con una presentación PowerPoint).
6. Evaluación y realimentación a los alumnos. La evaluación se hace en base al informe que los grupos presentan. La evaluación la lleva a cabo el profesor y es la misma para todos los miembros del equipo. La actividad tiene un peso del 20% de la calificación final de la asignatura.

Tabla 2. Diseño de la Actividad

Nombre	Problema: ¿Cómo evaluar un producto software contemporáneo?
Resultados de aprendizaje	<ol style="list-style-type: none"> 1. Identificar los contenidos de la asignatura con el tema tratado en el trabajo. 2. Encontrar fuentes válidas de información 3. Identificar soluciones a problemas basándose en la información disponible. 4. Argumentar en un debate la adecuación del modelo de calidad propuesto para resolver el problema de la evaluación de un producto software actual. 5. Reflexionar sobre la utilidad del modelo de calidad propuesto para tratar problemas sociale, éticos o medio ambientales.
Descripción	<p>La actividad consiste en i) desarrollar un modelo de calidad para solucionar el problema sobre cómo evaluar un tipo de producto software actual y ii) aplicar el modelo de calidad en la evaluación de un producto software concreto. La actividad involucra tres fases:</p> <ol style="list-style-type: none"> a) Entender el problema: elección del tema del trabajo, lectura, discusión en grupo, organizar las tareas y asignar responsabilidades; b) Búsqueda de información: los alumnos deben buscar información adicional sobre el dominio del problema para plantear soluciones creativas; c) Resolución del problema: puesta en común, presentación y discusión de resultados con el profesor; d) Evaluación del aprendizaje: los alumnos analizan el proceso para aprender. <p>No se han explicado las posibles soluciones al problema, se pretende que los alumnos reflexionen sobre ellas de antemano y planteen una solución.</p>
Evaluación	<ul style="list-style-type: none"> • <i>Redacción de informe.</i> Se ha desarrollado una plantilla para guiar a los alumnos en la elaboración del informe que recoge el planteamiento del modelo de calidad y su aplicación a un caso práctico. • <i>Rúbrica.</i> Se ha elaborado una rúbrica para evaluar la adecuación el modelo de calidad propuesto. • <i>Exposición oral.</i> Los alumnos exponen y discuten el trabajo en clase con el profesor y compañeros de clase.

En general, la actividad permite que el alumno se acerque a la realidad ya que realiza una actividad de carácter práctico cercana al trabajo que realizaría un ingeniero de calidad de software en un entorno profesional. Paralelamente trabajan otras competencias como pueden ser el trabajo en equipo y la comunicación efectiva.

2.3 Indicadores

Una vez se ha diseñado las actividades el desafío es identificar indicadores adecuados para medir el grado de adquisición de dicha competencia por parte de los alumnos. Se ha optado por desarrollar una rúbrica para el segundo nivel de dominio (3º y 4º de Grado) de la competencia general propuesta por la UPV (UPV 2015b) pero adaptando los indicadores a la propia naturaleza de la asignatura.

En particular, se ha adaptado todos los indicadores propuestos para este nivel de dominio a excepción del indicador “*Propone nuevas soluciones al problema que se ha planteado a partir de la propia experiencia y de la información disponible*” ya que se considera muy avanzado para el grado de madurez de los alumnos, el tipo de actividad que tienen que realizar y el tiempo que se dispone para trabajar la competencia..

La rúbrica corresponde al desarrollo del *proyecto* que involucra la definición y aplicación del modelo de calidad. Esta rúbrica permite al profesor evaluar el grado de adquisición de la competencia CT10 por parte de los alumnos. La rúbrica contiene 7 indicadores para evaluar la definición del modelo de calidad (Práctica 1), 6 indicadores para evaluar la aplicación del modelo a un caso práctico (Práctica 3) y otros 2 indicadores generales para evaluar la redacción del informe que recoge el desarrollo del trabajo, como se muestra en la Tabla 3. En el Apéndice A se presenta el instrumento completo junto con los cuatro niveles de consecución (“No alcanzado”, “En desarrollo”, “Bien/Adecuado” y “Excelente/Ejemplar”) para cada indicador, y se codifican como 1, 2, 3 y 4, respectivamente. La calificación final se hace como la suma de la puntuación obtenida en los diferentes indicadores y dimensiones contemplados.

Tabla 3. Indicadores propuestos para evaluar la CT10

Definición del modelo de calidad	Aplicación del modelo a un caso práctico
Identifica las causas que han llevado a la situación actual (necesidad del modelo)	Establece los requisitos de evaluación
Identifica las consecuencias o implicaciones sociales, económicas o culturales del problema en el contexto de la calidad del software	Especifica la evaluación
Distingue las partes que componen la calidad y relaciones entre ellas, identificando los principales actores afectados (descomposición de la calidad)	Proporciona escenarios de evaluación
Establece la importancia relativa de características de calidad	Describe aplicación de las métricas
Establece relaciones de impacto	Identifica defectos
Selecciona métricas de calidad	Razona (critica) la solución propuesta para resolver el problema
Establece criterios de decision	

2.4 Evaluación de la actividad

La última fase consiste en la evaluación de la actividad por parte de los alumnos a partir de un cuestionario de satisfacción anónimo. Esta actividad se ha llevado a cabo en clases de teoría. En base al cuestionario descrito en (VILLANUEVA et al., 2016), se ha definido un cuestionario donde el alumno valora el grado de acuerdo, utilizando una escala Likert de cinco niveles (totalmente de acuerdo – TA, más bien de acuerdo – MBA, término medio –

TM, más bien en desacuerdo – MBD, totalmente en desacuerdo – TD), con las siguientes afirmaciones/preguntas:

- P1. Creo que he aprendido más que si solo hubiera estudiado por mi cuenta estos contenidos.
- P2. El hecho de desarrollar un modelo de calidad como una solución práctica a la evaluación de un producto software contemporáneo potencia el aprendizaje de la asignatura.
- P3. Un producto software mal diseñado e implementado puede causar problemas técnicos, éticos, sociales o medio-ambientales.
- P4. Un modelo de calidad es un buen mecanismo para definir y evaluar cuestiones técnicas, éticas, sociales o medio-ambientales del uso del software en un dominio específico.
- P5. Los contenidos de la asignatura ayudan a resolver problemas como los presentados en la sesión de prácticas.

La primera pregunta se trata de una pregunta de control y se usa como diagnóstico de la asignatura. El resto de preguntas sirven para evaluar la opinión de los alumnos sobre de la utilidad de la actividad para su carrera profesional y el grado de integración de la competencia transversal “conocimiento de problemas contemporáneos” en la asignatura. Además, la encuesta también tenía una pregunta abierta que ha permitido a los alumnos expresar libremente sus opiniones sobre las actividades llevadas a cabo. El instrumento se ha implementado utilizando Google Forms. La recogida de datos se ha llevado a cabo al final del curso, al finalizar las sesiones presenciales.

4. Resultados

Esta sección describe los resultados de la evaluación del grado de adquisición de la competencia y la evaluación de la actividad por parte de los alumnos.

4.1 Evaluación del grado de adquisición de la competencia

Los profesores corrigen el informe y asignan una puntuación de 1 a 4 (No alcanzado, En desarrollo, Bien/adeecuado o Excelente) a cada uno de los 15 indicadores descritos en la rúbrica (ver Apéndice A) de acuerdo al nivel de desempeño de los alumnos.

La calificación final del trabajo se hace como la suma de la puntuación obtenida en los distintos indicadores y dimensiones contemplados. Por lo tanto, las medidas cualitativas han sido analizadas a través de la rúbrica que ha permitido producir una medida cuantitativa (nota del informe). Por último, se asigna un nivel de desempeño del grado de adquisición de la competencia a cada alumno mediante la escala descrita en la Tabla 4.

La Fig. 1 muestra los resultados de evaluación del grado de adquisición de la competencia transversal para el grupo, en función de los indicadores planteados en la rúbrica. Estos resultados indican que la mayoría de los alumnos (58 alumnos – 76%) han alcanzado la competencia completamente con un nivel de adquisición “B - Bien/Adeecuado”, seguido por un 7% que la han alcanzado excelentemente (5 alumnos).

Tabla 4. Calificación y equivalencia del nivel de adquisición de la competencia CT10

Calificación		
Puntos	Nota	CT10
50- 52	10	A
47 - 49	9,5	
44 - 46	9	
41- 43	8,5	
38 - 40	8	
35 -37	7,5	B
32 - 34	7	
29 - 31	6,5	
26 - 28	6	
25 - 27	5,5	
21 - 24	5	
17 - 20	4,5	C
13 - 16	4	
< =12	< 4	D

Por otra parte, un 8% (6 alumnos) la han alcanzado parcialmente con un nivel de adquisición “C – en Desarrollo” y solo un 3% (2 alumnos) no han alcanzado la competencia. Un 7% (5 alumnos) no han realizado el trabajo académico. En general, estos resultados son positivos ya que indican que la mayoría de los alumnos de este grupo han adquirido la competencia.

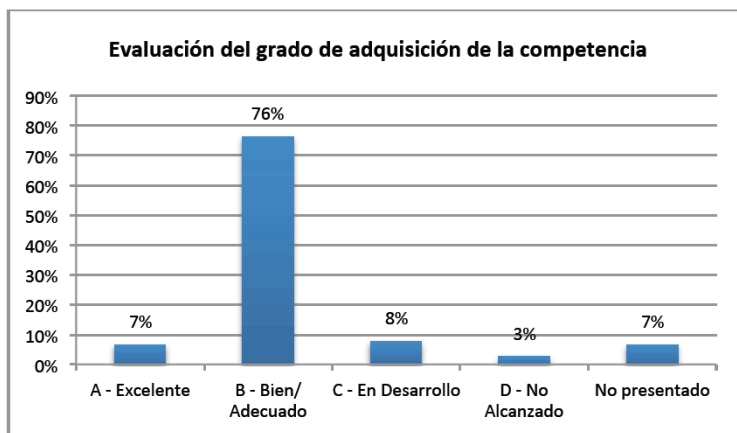


Fig 1. Resultados del grado de adquisición de la competencia según la escala utilizada en la rúbrica del Apéndice A

4.2 Evaluación por parte de los alumnos

El cuestionario de satisfacción ha sido respondido por 71 alumnos. La Fig. 2 muestra los resultados obtenidos en el que se representa los valores del número de respuestas para cada pregunta. Los resultados son muy favorables ya que la mayoría de los alumnos están más bien de acuerdo o totalmente de acuerdo con las distintas afirmaciones.

Las respuestas a la pregunta abierta del cuestionario han mostrado que los alumnos han valorado positivamente las actividades ya que les permiten relacionar los contenidos teóricos de la asignatura con una actividad que realizarían en su entorno profesional.

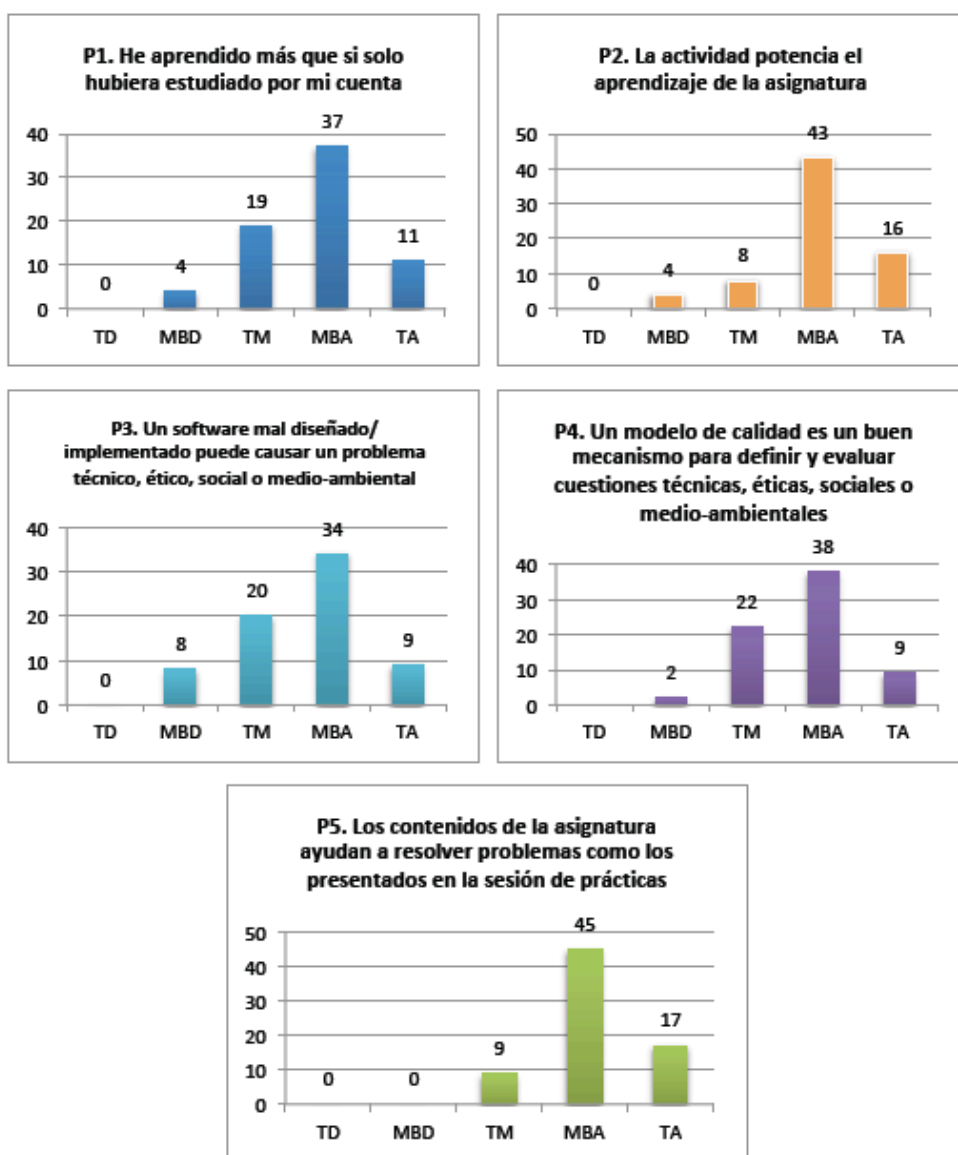


Fig 2. Resultados del cuestionario de evaluación de la actividad

5. Conclusiones

En este trabajo se ha presentado la integración de la competencia transversal de “conocimientos de problemas contemporáneos” en una asignatura del grado de Ingeniería Informática de la UPV. Se ha definido la competencia en el contexto de la asignatura mediante la definición de actividades formativas, resultados de aprendizaje, indicadores y una rúbrica que permite medir el grado de adquisición de la competencia.

Se ha presentado los resultados de la experiencia piloto realizada en 2017. Las actividades han permitido que los alumnos se acerquen a la realidad del sector en el que se están formando a través del desarrollo de un modelo de calidad, alineado con los principales

estándares, y dotado de atributos de calidad y métricas que permiten caracterizar y medir los aspectos éticos, sociales o medioambientales de los sistemas software.

Los resultados de la rúbrica han indicado que las actividades realizadas han apoyado el desarrollo de la competencia. En particular, los alumnos han aprendido a discriminar fuentes de información, a detectar problemas potenciales en los sistemas software y a proponer mecanismos para evaluarlo. El seminario realizado ha permitido debatir estos problemas en clase y discutir el grado de bondad de las soluciones planteadas. El seminario también ha permitido que los alumnos conozcan otros ejemplos expuestos por sus compañeros y se familiaricen con los problemas existentes en la evaluación de sistemas software contemporáneos. Además, los resultados de la encuesta realizada para medir la utilidad de las actividades indican que los alumnos han valorado muy bien las actividades realizadas y las consideran útiles para su desarrollo profesional.

Como planteamiento de mejora para los próximos años, se pretende realizar un análisis más profundo sobre qué aspectos sociales, éticos o medioambientales concretos son tratados en los distintos trabajos realizados por los alumnos. Otro aspecto de mejora sería utilizar un cuestionario estándar (o desarrollar uno propio) para la autoevaluación del nivel de la competencia por parte de los alumnos. De esta forma, podría utilizarse un pre- y post-test para hacer un análisis pareado de los datos y determinar, realmente, si la actividad ha contribuido a la mejora del grado de adquisición de la competencia. También se pretende valorar la incorporación de otras acciones formativas como visitas externas o charlas con profesionales. Por último, se pretende analizar los datos obtenidos durante el curso 2018 y compararlo con los resultados obtenidos en esta experiencia piloto.

6. Referencias

FERNÁNDEZ MARCH A (2006). “Metodologías activas para la formación en competencias”. *Education Siglo XXI*, 24. pp 35-56.

OLTRA-BADENES R., GIL-GÓMEZ H (2017). “Innovación docente para el desarrollo de la competencia transversal “Conocimiento de problemas contemporáneos” en el marco del proyecto de competencias transversales de la UPV”, en Congreso Nacional de Innovación Educativa y de Docencia en Red (In-Red 2017), UPV, 13 y 14 de julio de 2017.

ISO/IEC 25010:2011, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuARE) -- System and software quality models.

ISO/IEC 25040:2011, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuARE) -- Evaluation process.

Universitat Politècnica de València, UPV (2015a). Presentación de Competencias Transversales UPV, CT-10. Conocimiento de problemas contemporáneos.

Universitat Politècnica de València, UPV (2015b). Rúbrica UPV CT-10. Conocimiento de problemas contemporáneos.

VILLANUEVA J.F., BARRACHINA T., GALLARDO S., LAJARA-CAMILLERID N. y ROVIRE A. “Integración de la competencia transversal de “conocimiento de problemas contemporáneos” en Asignaturas de Grado”, en Congreso Nacional de Innovación Educativa y de Docencia en Red (In-Red 2016). Valencia, 7 y 8 de julio de 2016.

World Business Council for Sustainable Development’s Vision 2050, WBCSD (2019). <https://www.wbcsd.org/Overview/About-us/Vision2050>. [Consulta: 15 de febrero de 2019]

Apéndice A – Rúbrica

INDICADORES	4 (Excelente/ejemplar)	3 (Bien/adecuado)	2 (En desarrollo)	1 (No alcanzado)
Definición del Modelo de Calidad				
Identifica las causas que han llevado a la situación actual (necesidad del modelo)	Identifica muchas causas y las explica detalladamente	Identifica pocas causas y no las explica	Identifica causas suficientes y las explica parcialmente	No identifica ninguna causa
Identifica las consecuencias o implicaciones sociales, económicas o culturales del problema en el contexto de la calidad del software	Identifica muchas consecuencias y las explica detalladamente	Identifica las consecuencias y las explica parcialmente	Identifica pocas consecuencias y no las explica	No identifica ninguna consecuencia
Distingue las partes que componen la calidad y relaciones entre ellas, identificando los principales actores afectados (Descomposición de la calidad)	Descompone el modelo de calidad de forma clara, correcta y en estrecha relación con los intereses de los actores y el dominio seleccionado	Descompone el modelo de calidad de forma clara y en estrecha relación con los intereses de los actores y dominio seleccionado pero presenta algunos errores	La descomposición del modelo presenta algunos errores y además algunas características/atributos no guardan relación con los actores o dominio seleccionado	La descomposición del modelo presenta varios errores y no guarda relación con los actores o dominio seleccionado
Establece la importancia relativa de características de calidad	Proporciona argumentos convincentes para la importancia relativa de todas las características de calidad	Proporciona argumentos convincentes para la importancia relativa de la mayoría de características de calidad	No se proporciona argumentos convincentes para la importancia relativa de algunas características del modelo	No se proporciona argumentos convincentes para la importancia relativa de la mayoría de características del modelo
Establece relaciones de impacto	Identifica al menos 4 relaciones de impacto entre atributos de calidad y todas son correctas y pertinentes	Identifica al menos 3 relaciones de impacto entre atributos de calidad y todas son correctas y pertinentes.	Identifica al menos 2 relaciones de impacto entre atributos y todas son correctas y pertinentes.	No se identifica las relaciones de impacto entre atributos de calidad o casi todas son incorrectas
Selecciona métricas de calidad	Identifica muchas métricas para medir los atributos de calidad y todas son muy adecuadas	Identifica suficiente métricas para medir los atributos de calidad y todas son adecuadas	Identifica algunas métricas para medir los atributos de calidad pero hay algunos errores.	No se identifican suficiente métricas o éstas no son adecuadas para medir los atributos de calidad
Establece criterios de decisión	Define corretamente criterios de decisión y umbrales para todas las métricas del modelo	Define corretamente criterios de decisión y umbrales para la mayoría de métricas del modelo	Define corretamente criterios de decisión y umbrales para algunas métricas del modelo	No se define criterios de decisión y umbrales para las métricas o los criterios son incorrectos

Aplicación del modelo de calidad a un caso práctico				
Establece los requisitos de evaluación	Describe los requisitos de evaluación (objetivo, artefacto a evaluar, atributos de calidad, etc.) de forma clara y objetiva	Los requisitos de evaluación (objetivo, artefacto a evaluar, atributos de calidad, etc.) carecen de alguna información relevante	Se observan algunos errores en los requisitos de evaluación	No se describe los requisitos de evaluación o éstos son confusos
Especifica la evaluación	La especificación de la evaluación (selección de métricas, operacionalización de métricas y umbrales) se describe de manera clara y objetiva	La especificación de la evaluación (selección de métricas, operacionalización de métricas y umbrales) carece de alguna información relevante	Se observan algunos errores en la especificación de la evaluación	La especificación de la evaluación (selección de métricas, operacionalización de métricas y umbrales) es confusa y carece de información relevante
Proporciona escenarios de evaluación	Describe los escenarios (páginas web, IU, código fuente, etc.) donde se aplican todas las métricas de forma clara y objetiva	Describe los escenarios (páginas web, IU, código fuente, etc.) donde se aplican casi todas las métricas de forma clara y objetiva	No se describe los escenarios (páginas web, IU, código fuente, etc.) donde se aplican algunas métricas	No se describe los escenarios donde se aplican todas las métricas
Describe la aplicación de las métricas	Describe la medición de todas las métricas de forma clara y objetiva	Describe la medición de la mayoría de las métricas de forma clara y objetiva	La aplicación de algunas métricas es confusa y difícil de entender	No se describe la aplicación de las métricas o ésta es confusa y difícil de entender
Identifica defectos	Identifica 4 o más defectos	Identifica 3 defectos	Identifica 2 defectos	Identifica un solo defecto
Razona (critica) la solución propuesta para resolver el problema	Razona/critica la solución propuesta y la analiza de forma óptima/satisfactoria	Razona/critica la solución propuesta de forma suficiente	Razona/critica la solución propuesta de forma incompleta	No razona/critica la solución propuesta o lo hace de forma errónea