

# La experiencia de un docente que hace no tanto era alumno

Jose Divasón  
Departamento de Matemáticas y Computación  
Universidad de La Rioja  
26006 Logroño  
jose.divasonm@unirioja.es

## Resumen

En este trabajo presento cómo ha sido mi primera experiencia docente en el marco universitario, sobre todo teniendo en cuenta que quería aprovechar la ventaja de haber estudiado en la misma universidad en la que ahora imparto docencia. Los cambios introducidos en la asignatura de Sistemas Distribuidos se basan tanto en mi opinión de estudiante de años atrás como en las opiniones del alumnado de este año. Se muestra desde los primeros pasos recabando las ideas de los alumnos, hasta los cambios introducidos y los problemas surgidos de cambiar una asignatura sobre la marcha.

## Abstract

In this work I present my first teaching experience in a university context, taking into account that I aimed at making the best of having studied in the same university where I am now a lecturer. The changes that have been introduced in the subject of Distributed Systems are based on both my experience as a student and the feedback obtained from the current students. The first changes, the opinions of the students and the problems that have arisen from modifying a subject on its way are also shown.

## Palabras clave

Experiencia docente, Sistemas Distribuidos, gamificación.

## 1. Introducción y motivación

Durante el primer cuatrimestre del curso 2016/2017 he sido por primera vez docente universitario de una materia completa, concretamente de la asignatura de Sistemas Distribuidos en la Universidad de La Rioja. Es importante remarcar que finalicé mis estudios de ingeniería en informática en la misma universidad en el

año 2011. Por tanto, conocía de primera mano la universidad, el estilo de cada profesor, los contenidos, y sobre todo tenía mi opinión (como estudiante) acerca de las asignaturas.

Consideraba que esto debía ser una ventaja, ya que el hecho de tener muy reciente mi etapa como estudiante me podía servir para intentar mejorar la docencia de la materia. En 2011 nada más acabar la carrera, tenía mi propia opinión personal (y quizás errónea) como estudiante acerca de algunas asignaturas:

- A veces se incidía mucho en la teoría, que se olvidaba rápido, podía quedar desactualizada al cabo de pocos años por el avance de la informática.
- Algunos contenidos no parecían ser de aplicación directa en el plano profesional.
- A veces las prácticas eran muy guiadas y no resolvían problemas reales, cuando llegase a empresa estaría perdido.
- Los contenidos me parecían interesantes, pero a veces las asignaturas no conseguían motivarme totalmente.

Cabe destacar también que a lo que me enfrentaba era a mi primera experiencia docente en el marco universitario, y concretamente iba a ser con la asignatura de Sistemas Distribuidos. Hasta entonces estaba muy acostumbrado a los congresos científicos y como es obvio son ambientes muy distintos. Sin embargo, creía que era algo que también podía resultarme útil. Stephanie Weirich tiene unas diapositivas [6] en las que da unos consejos sobre cómo dar una buena charla de investigación. En mis años de doctorado siempre los había seguido y me parecía que muchas cosas coincidían para aplicarlas en una clase. Nos gustaría tener unos estudiantes que dominen los conceptos previos de la asignatura, la lleven al día, que estén motivados, frescos, en alerta y listos para la acción. Salvo excepciones, la realidad no es esa. Weirich afirma que en una charla de investigación el ponente tiene 2 minutos para que la audiencia se enganche al tema antes de que se aburra y decida desconectar. Por tanto, la misión de un ponente es despertar a la audiencia y que ésta acabe contenta

de no haber desconectado. Es decir, en el fondo el consejo es motivar lo antes posible. Me parecía una idea aplicable a clase, y más teniendo en cuenta que las clases estaban programadas a primera hora de la tarde. Mi opinión (porque me pasaba a mí) es que a veces los alumnos no ven determinadas asignaturas como algo interesante. Claro que saben que son muy importantes para ellos, pero no interesantes. Y de ahí que a veces se pregunten (yo mismo cuando era alumno lo hacía), ¿y esto realmente sirve para algo y se utiliza en el plano profesional?

Quería que mis alumnos no se hiciesen esa pregunta. Un reto importante que tenía era que debía enfrentarme a una asignatura sin apenas tiempo y mejorarla sobre la marcha. Además, no siempre uno puede hacer todos los cambios que desearía por diversos motivos, como restricciones impuestas por las normativas o software no instalado en las aulas de laboratorio.

Hay multitud de nuevas técnicas para hacer las clases más dinámicas, como por ejemplo el aprendizaje basado en problemas o la metodología de la clase inversa. De hecho se han presentado varias experiencias usando estas técnicas con resultados muy positivos en el contexto del Grado en Ingeniería Informática [4, 5]. Estas formas de enseñanza que me parecían muy interesantes y me gustaría aplicarlas al menos en una parte de la asignatura, pero me parecía totalmente inviable por el tiempo que tenía para preparar las cosas.

En los apartados que siguen a continuación se muestran los antecedentes de la asignatura (Apartado 2), las estrategias, objetivos y cambios propuestos (Apartado 3), las experiencias con los trabajos (Apartado 4) y la gamificación en el aula (Apartado 5). Finalmente las conclusiones se muestran en el Apartado 6.

## 2. Sistemas Distribuidos

Sistemas Distribuidos es una asignatura obligatoria de tercero del Grado en Ingeniería Informática. En la Universidad de La Rioja es una asignatura programada para ser “eminente práctica”, pero tradicionalmente considerada como una asignatura “hueso” por los estudiantes. Tiene 6 créditos, de los cuales 3,2 son de teoría y 2,8 de prácticas de laboratorio en las que se programa en Java. Las clases teóricas eran totalmente “lección magistral”, mientras que en las de laboratorio los alumnos resolvían problemas de programación y pequeños proyectos donde se aplicaban los contenidos teóricos de la asignatura.

Como la Universidad de La Rioja es una universidad *pequeña*, el número de alumnos matriculados en la asignatura de Sistemas Distribuidos no es muy grande, suele estar entre 25 y 30 alumnos. El Cuadro 1 muestra la información acerca de los estudiantes matriculados y aprobados (en la convocatoria de enero) del curso

### Comparte tu manera de innovar: aprendamos juntos

	Año (conv. enero)		
	2016	2015	2014
<b>Matriculados</b>	29	25	25
<b>Aprobados</b>	24	16	8

Cuadro 1: Numero de matriculados y aprobados en la convocatoria de enero de Sistemas Distribuidos en cursos anteriores.

2016/2017 y los tres cursos anteriores.

El sistema de evaluación en años anteriores consistía en un examen final (60 %), prácticas de laboratorio entregables (30 %) y trabajos (10 %). El examen final constaba tanto de preguntas teóricas como de preguntas prácticas. Estas últimas eran preguntas de desarrollo de código que debían responderse en papel. Las prácticas de laboratorio se entregaban cada 2 semanas y no eran recuperables. Como criterio de evaluación, se requería aprobar tanto el examen final como las prácticas de laboratorio para poder aprobar la asignatura. Los trabajos eran optativos y no recuperables, e incluso se podía obtener puntos extras por la realización de mejoras en ellos, superando ese 10 % de la calificación final. Por normativa, el sistema de evaluación para el curso 2016/2017 se mantuvo igual, aunque me hubiese gustado disminuir el peso del examen final en favor de las prácticas y trabajos.

## 3. Objetivos, estrategias y cambios

En mi primer día como docente, ofrecí a los 30 estudiantes matriculados de Sistemas Distribuidos algo muy simple pero que no habían tenido la oportunidad de hacer anteriormente: la posibilidad de expresar por escrito anónimamente lo que quisieran sobre la asignatura antes de empezar (qué esperaban, qué contenidos les gustaría ver con detalle...). En el fondo no buscaba obtener comentarios concretos para la asignatura de Sistemas Distribuidos, sino conocer qué opinaban de su educación en la universidad hasta el momento. Hubo multitud de comentarios que coincidían con mi opinión de años atrás: “aprender nuevos conceptos prácticos para ser capaces de desarrollar un trabajo fin de grado y otros proyectos en empresa en el futuro”, “tener la sensación de que he aprendido algo útil para mi futuro laboral, no un contenido teórico que se me va a olvidar antes de acabar la carrera”. Los estudiantes también remarcaban que los contenidos no solo fuesen prácticos, sino aplicables a proyectos reales. Mi opinión también era esa: las prácticas y los trabajos no deberían ser tareas aisladas a las que se les enmascara un contexto más atractivo para intentar hacerlas más interesantes, sino que a partir de un objetivo o problema

	2014	2015	2016	2017
<b>Trabajo 1</b>	60 %	60 %	86 %	90 %
<b>Trabajo 2</b>	67 %	55 %	71 %	62 %

Cuadro 2: Porcentaje de estudiantes que hicieron los trabajos optativos.

real existente, intentar solucionarlo con los contenidos de la asignatura.

En definitiva, me di cuenta de que las inquietudes y sensaciones que tenía 5 años atrás como estudiante mayormente se mantenían en los alumnos actuales. Se marcaron los siguientes objetivos y estrategias:

- Aumentar el interés por la asignatura y favorecer la participación.
- Actualización de contenidos.
- Realización de trabajos eminentemente prácticos que les resultasen interesantes.
- Gamificación.

El aumento del interés se centró en intentar lograr aumentar la motivación general del aula, de forma que los alumnos se diesen cuenta de que los contenidos que se impartían en la asignatura eran de uso real en el plano profesional y que gracias a ellos iban a ser capaces de resolver problemas reales. Para ello con cada concepto teórico y práctico se mostraban ejemplos de usos en casos reales. Para aumentar la participación intenté que las clases fuesen más dinámicas, y que los propios estudiantes pensasen, mostrasen y defendiesen delante de toda la clase sus soluciones a pequeños problemas propuestos tras cada tema. Otras veces dichas soluciones se debían pensar en grupo y resolver colaborativamente. También se actualizaron determinados contenidos de la asignatura, por ejemplo nos centramos mucho más en servicios web de tipo API Restful (más actual y usado) que en SOAP (que está quedando obsoleto y en desuso). La experiencia con los trabajos y la gamificación en el aula fueron los cambios más notables que se realizaron en la asignatura y se presentan con detalle en las secciones 4 y 5.

Al ser un grupo de alumnos pequeño, se pudo hacer algo muy importante y que los estudiantes valoraron muy positivamente: las prácticas de laboratorio fueron corregidas tras cada entrega y devueltas a cada uno para darles un feedback de sus fallos y cómo llevan la asignatura.

## 4. Los trabajos

Los trabajos son una parte de la evaluación de la asignatura. Su realización era optativa, y por tanto no había ningún requisito de obtener nota mínima para poder aprobar la asignatura. Durante todos los años anteriores se habían propuesto un mínimo de 2 trabajos.

Un primero algo más teórico y sencillo; y un segundo más largo con más programación. A veces incluso un tercero que extendiese al segundo, fundamentalmente pensado para los alumnos que quisieran subir nota. Por ejemplo, en el curso 2015/2016 se propusieron los siguientes:

1. Trabajo teórico de búsqueda de información sobre cómo incorporar un TPV Virtual en una aplicación web para permitir el pago electrónico.
2. Diccionario de términos informáticos como servicio web.

Teniendo en cuenta la opinión que tenía de mis años de estudiante y la información que había recopilado a través de los comentarios que obtuve de los alumnos, para el curso 2016/2017 decidí cambiar totalmente los trabajos para hacerlos mucho más prácticos y sobre todo para conseguir que los estudiantes viesan que iban a desarrollar algo que realmente era atractivo o que iba a tener una utilidad real.

El primer trabajo consistió en implementar un programa para detectar similitudes entre ficheros de texto, con el trasfondo de intentar aplicarlo en la detección de copias. Es importante remarcar que en reuniones de coordinación de años anteriores se detectó que había habido copias de prácticas entre los estudiantes.

Se propuso partir de los conocimientos adquiridos en la primera parte de la asignatura y de una librería ya existente que calculaba diferencias de ficheros en Java. La propuesta concreta consistía en diseñar e implementar un programa que dado un conjunto de ficheros (más concretamente las prácticas descargadas a través del aula virtual de la asignatura), normalizase los ficheros (eliminando las líneas en blanco, los espacios al principio y al final de cada línea, eliminando los comentarios, ...) y calculase porcentajes de similitud entre ficheros según distintas medidas (porcentaje de líneas de un fichero que estaban en otro y distancia de Levenshtein), avisando al usuario en caso de que haya ficheros con una similitud muy alta. Realmente ya existen varios programas mucho mejores y especializados en la detección de copias de software, pero la idea era que programasen ellos desde cero algo que realmente fuese usable y útil.

El trabajo valía 0,5 puntos (un 5 % de la nota final de la asignatura) si cumplía las funcionalidades básicas requeridas, pero se podían obtener hasta 0,9 puntos si se implementaban mejoras (como por ejemplo, mejores medidas, detectar reemplazos, intercambios, hacerlo como servicio web...). Los estudiantes tuvieron casi un mes para hacerlo.

Al ser un trabajo no obligatorio y que "solo" valía 0,5 puntos del total de la asignatura, los requisitos básicos estaban acordes a la nota. El objetivo del trabajo no era programar software completo y totalmente específico para solucionar dicha tarea, de hecho podría

ser mejorado en muchos sentidos (hacerlo específico para un lenguaje de programación en concreto, mejores medidas, uso machine learning, etc...). El objetivo era que los estudiantes pusiesen en práctica los conocimientos adquiridos y con ellos se viesen capaces de implementar algo que con utilidad práctica real. Para remarcar su usabilidad real, uno de los requisitos era que fuese muy eficiente (mediante el uso de estructuras de datos adecuadas, minimizando el acceso a ficheros, concurrencia, hilos en paralelo o incluso repartir el trabajo en varios ordenadores). Este requisito fue valorado positivamente por los estudiantes, ya que de normal hasta entonces no estaban acostumbrados a cuidar la eficiencia. En la reunión de coordinación de este año se constató que no se habían detectado copias en ninguna asignatura de tercero, por tanto, no se puede garantizar pero quizás también sirvió como elemento de disuasión para realizar copias.

Realmente el programa desarrollado es simple, ya que es un programa que calcula diferencias de ficheros, el porcentaje de líneas completas de un fichero que están en también en otro y porcentajes a partir de la distancia de Levenshtein entre el contenido de los ficheros. No tiene elementos avanzados para detectar copias (como por ejemplo serían detectar intercambios, reemplazos de variables, etc...), pero al menos puede servir de ayuda para detectar de forma muy rápida plagios obvios como en el caso de ficheros o partes de ficheros prácticamente iguales.

El programa también lo utilicé en el segundo cuatrimestre con las prácticas de otra asignatura y me sirvió para detectar dos plagios. También es cierto que detecté varias más pero no a través del programa, puesto que eran esencialmente el mismo código con los nombres de las variables distintas, y eso el programa no lo detecta. A partir de las opiniones anónimas recopiladas, los estudiantes valoraron muy positivamente el trabajo y la participación fue muy alta: el 90 % de los estudiantes lo realizaron, lo cual supuso una mejora con respecto a años anteriores (ver Cuadro 2) a pesar de que era un trabajo más difícil. Principalmente remarcaban que era un trabajo que les parecía interesante porque programaban desde cero y a partir de los conocimientos adquiridos en la asignatura una aplicación real. Muchos reseñaron que el hecho de exigir eficiencia había sido un reto muy importante que les había obligado a pensar y a realizar muchas pruebas de rendimiento.

El segundo trabajo puntuaba hasta un 10 %, y consistió en el diseño e implementación de un sistema de intercambio de archivos P2P centralizado con un servidor que almacenase las IPs, puertos e información de los ficheros de los nodos conectados en cada momento. El objetivo era poner en práctica y profundizar en algunos de los conocimientos aprendidos en la asignatura sobre las APIs de entrada/salida de Java, los sockets y

la programación concurrente mediante el desarrollo de un programa desde cero que resultase atractivo.

Los estudiantes debían implementar tanto los clientes (los nodos), como el servidor de directorio (el servidor central). No era un trabajo guiado: cada estudiante diseñaba e implementaba como quería sabiendo los requisitos que tenía que cumplir el programa. Los estudiantes tuvieron 5 semanas para realizarlo.

Fue un trabajo completo porque permitía practicar muchos de los contenidos de la asignatura: trabajo con sockets, trabajo con planificadores de tareas, elección de estructuras de datos adecuadas, paso de mensajes, diseño e implementación de un protocolo básico, problemas de concurrencia, envío de varios archivos a la vez, recepción de un archivo troceado en varias partes a partir de varios nodos, trabajo con funciones hash para comprobar la integridad de los archivos recibidos y distinguir los ficheros (nodos distintos pueden tener un mismo fichero pero con nombres diferentes, nodos distintos podrían tener dos ficheros distintos con el mismo nombre), etc...

Todos los conocimientos necesarios para realizar los trabajos habían sido impartidos entre las horas de teoría y practicados en las clases de laboratorio. Los estudiantes mostraron dos comentarios: el tema del trabajo les resultaba muy atractivo y motivador, pero a una parte de la clase le parecía un trabajo difícil. En cualquier caso el trabajo fue entregado completo por el 62 % de la clase, cifra similar a la media de los años anteriores.

## 5. Gamificación

La gamificación es un concepto que está ganando adeptos incluso en el marco de la docencia universitaria. Consiste en el empleo de técnicas y dinámicas propias de los juegos para incentivar la participación y motivación del alumnado, de modo que se fomente un determinado comportamiento deseado. Varias experiencias positivas ya se han presentado en las JENUJ de años anteriores [1, 3].

Recordaba mi época de alumno y nunca había tenido ninguna experiencia así, pero consideraba que alguna vez me hubiese venido bien, sobre para mantener la motivación y tener un reto o una competición por delante. Así que decidí llevar a cabo un pequeño juego en mi clase de Sistemas Distribuidos.

Fue algo fácil de preparar: un concurso de preguntas y respuestas por equipos sobre los contenidos de la asignatura. Los objetivos eran claros, quería proponer un reto de forma que se motivase el estudio, se buscaba fomentar la participación, la resolución de dudas y el repaso global antes del examen puesto que se llevaría a cabo el último día de clase.

La metodología era sencilla: se distribuía la clase en equipos de 3 o 4 personas repartidas equitativamente

## PREGUNTA

Queremos contar todas las líneas de los ficheros de un directorio concurrentemente. Supongamos que:

- Tenemos creado un pool y el directorio.
- No nos preocupamos de si todo lo de dentro son ficheros de texto
- Tenemos la clase adecuada que cuenta las líneas.

¿Es correcto ejecutarlo así? ¿Por qué?

```
int lineas = 0;
for (File f : directorio.listFiles())
{
    Callable<Integer> r = new CuentaLineasCallable(f);
    lineas += pool.submit(r).get();
}
```

## RESPUESTA

No es correcto ejecutarlo así. En el fondo estamos ejecutando secuencialmente: en cada iteración creamos la tarea (con `pool.submit(r)`) y esperamos a que termine (con el `.get()`). Es más, será más lento que secuencialmente porque metemos la sobrecarga del pool y de crear hilos.

La solución es crear dos bucles: uno para meter los objetos en el pool y almacenar los `Future<Integer>` en una lista, y después otro para procesar los resultados de esa lista.

```
int lineas = 0;
List<Future<Integer>> list = new ArrayList<Future<Integer>>();
for (File f : directorio.listFiles())
{
    Callable<Integer> r = new CuentaLineasCallable(f);
    list.add(pool.submit(r));
}
int suma = 0;
for (Future<Integer> resultadoParcial : list)
{
    suma += resultadoParcial.get();
}
```

Figura 1: Una de las preguntas del concurso y su respuesta.

te según las notas obtenidas hasta el momento en las prácticas para evitar tener equipos descompensados. Se mostraba una pregunta en el proyector y uno de los equipos tenía 2 minutos para responderla. Las preguntas podían ser de cualquier concepto de la asignatura, incluso de código, pero no se podía usar el ordenador para resolverlas. Tenían que ser factibles de contestar en un minuto (pero había 2 minutos de tiempo para contestarlas) y no debían ser preguntas en las que la posible respuesta esté entre 2 opciones (porque la idea era que si un equipo no acierta, otro equipo tenga la posibilidad de responder).

Se otorgaban puntos por respuesta correcta según el tiempo empleado y por respuesta correcta tras un rebote (es decir, cuando otro equipo ha fallado). Para fomentar el estudio no se permitía el uso de ningún elemento tecnológico ni los apuntes.

Previo al concurso, los propios grupos de estudiantes eran los que prepararon las preguntas y la explicación con la respuesta detallada correcta. De esta forma los alumnos estudiaban, veían que conceptos o dificultades se habían encontrado, lo plasmaban en una pregunta para el resto de equipos del concurso y además dejaban la solución completa escrita para que sirviese de repaso posterior. Además, en caso de que un equipo propusiese una pregunta (siempre acorde al nivel de los contenidos de la asignatura) y ninguno de los otros equipos fuese capaz de responderla, también se valoró con puntos extras.

Los alumnos me enviaban todas las preguntas con sus correspondientes respuestas explicativas detalladas y yo me encargué de recopilarlas en un documento PowerPoint para usarlo en clase, añadiendo el contador de tiempo límite para responder. Tras el concurso, dicha recopilación de preguntas y respuestas fue publicada en el aula virtual de la asignatura, para que sirviese como repaso al examen. Como motivación para repa-

sarlas aún más, avisé de que alguna de las preguntas caería en el examen.

La experiencia fue acogida positivamente por los estudiantes, todos ellos participaron y se recopilaron más de 40 preguntas y respuestas. La Figura 1 muestra un ejemplo de una de las preguntas recopiladas y su respuesta.

## 6. Resultados, conclusiones y trabajo futuro

En este artículo se ha mostrado la primera experiencia docente universitaria del autor, explicando los cambios introducidos en la asignatura de Sistemas Distribuidos a partir de la opinión y el punto de vista que tenía como estudiante años atrás, además de la información recopilada de los alumnos.

El principal objetivo era aumentar tanto la participación como la motivación por la asignatura, haciendo ver a los alumnos que no solo era una asignatura importante, sino que además vieses que era útil. Además de una actualización de contenidos y el fomento de la participación en clase, el principal cambio fue la realización de prácticas y trabajos que resultasen más atractivos a los estudiantes, además de una pequeña actividad de gamificación.

Cabe resaltar de nuevo que se fomentó muchísimo la participación en el aula intentando hacer las clases más dinámicas. La experiencia me mostró que los alumnos no son reacios a participar, preguntar, responder o salir a resolver ejercicios. Está claro que el contexto universitario históricamente ha tendido mucho a la clase magistral, pero me sorprendió mucho oír a algún estudiante decir “es la primera vez en tres años que salgo a la pizarra”.

Las opiniones de los estudiantes que se han ido mos-

trando a lo largo del artículo se recabaron a través de hojas en blanco anónimas (para que pudiesen expresar libremente sus opiniones) a principio, mitad y final de la asignatura. El Cuadro 3 también muestra los resultados obtenidos a partir de las encuestas de la propia universidad. Los alumnos han visto la asignatura como más interesante y la han valorado mejor que, en media, el resto de asignaturas del primer cuatrimestre de todo el Grado en Ingeniería Informática, tanto comparando en este curso como en los tres anteriores. No se tiene acceso a los datos concretos de Sistemas Distribuidos en años anteriores por ser datos privados de distinto profesorado.

El examen final valía un 60 % de la asignatura. Siempre había tenido una parte más bien teórica y otra parte de problemas parecidos a los propuestos en las prácticas de laboratorio, pero esta vez para resolver en papel. Para el curso 2016/2017, la puntuación del examen no se podía cambiar por motivos de plazos y normativas. El examen pasó a ser eminentemente práctico, con tan solo una pregunta teórica pura que valía 1 punto sobre 10 del examen. El resto fueron o bien problemas prácticos o bien preguntas cortas de código de conceptos de la asignatura.

El número de matriculados en la asignatura era de 30, de los cuales 29 asistieron con total regularidad a las clases de teoría y de prácticas (la otra persona matriculada no asistió a ninguna de las clases desde el inicio del curso). La totalidad de esos 29 alumnos se presentaron al examen, es decir, se consiguió que no hubiese abandonos de la asignatura a lo largo del curso. En cuanto al número de aprobados en la convocatoria de enero, fue de 22 sobre los 29 presentados. Considero que es un dato importante pero que no se debería comparar y sacar conclusiones a partir de los datos de años anteriores. Comparar el número de aprobados y suspensos con respecto a años anteriores podría no ser fiable, ya que ha habido distintos profesores, distinta dificultad de trabajos, distintas prácticas, distintos exámenes y distintos estudiantes. Incluso podría darse el caso de que un alumno haya llegado a superar una asignatura sin haber dominado sus conceptos básicos, cosa que debería evitarse a toda costa. En cualquier caso, las opiniones recabadas de los estudiantes (como ya se ha dicho, a través de hojas en blanco anónimas para poder hacer comentarios libremente) fueron muy positivas tanto en cuanto a los contenidos vistos como en la importancia y utilidad que le veían a la asignatura, lo cual concuerda con los datos obtenidos a través de las encuestas de la propia universidad (ver de nuevo Cuadro 3).

Como reflexión final, es innegable decir que un docente con experiencia imparte mucho mejor las clases. En mi caso, no tenía experiencia impartiendo clase, pero sí mucha experiencia como alumno, así que inten-

té aprovecharme de ello. El factor edad podría incluso verse como una ventaja: indudablemente podía conocer de un modo más cercano las inquietudes y la visión que puede tener un estudiante de esta asignatura, y la intenté adaptar adecuadamente sin dejar de lado sus contenidos y su exigencia.

La metodología aplicada ha sido posible debido al bajo número de alumnos (30 matriculados). Como desventaja hay que decir que requiere bastante dedicación por parte del profesor, por ejemplo, todas las prácticas eran corregidas tras cada entrega y mostradas a los estudiantes para que aprendiesen de sus errores. Hubiese sido muy deseable tener por ejemplo alguna herramienta de generación automática de pruebas de validación para facilitar dicha tarea. Ya hay experiencias en este sentido en otras universidades, incluso centradas en la asignatura de Sistemas Distribuidos y sus prácticas de laboratorio [2]. Las tutorías para resolver dudas de los trabajos, especialmente el del sistema de intercambio de archivos P2P, fueron también muy numerosas y constantes a lo largo de todo el cuatrimestre.

En cualquier caso, considero que la experiencia ha sido muy positiva y los estudiantes han acabado contentos. Con respecto a posibles cambios para años posteriores a partir de lo aprendido este año, muchos estudiantes no veían adecuados los criterios de evaluación fijados, sobre todo el peso del examen. Coincido con ellos, al ser una asignatura tan práctica creo que lo más importante sería rebajar el peso del examen en la calificación final. A pesar de ser un examen centrado en los contenidos prácticos, tener que escribir código en papel es pesado. Quizás sería más recomendable hacer dos exámenes: uno en el aula de laboratorio y otro en el aula de teoría. En cualquier caso, reduciría su peso en la calificación final en favor de las prácticas y los trabajos, que además podrían convertirse en proyectos un poco más grandes y completos. Como posibles mejoras para cursos posteriores está también el uso de la clase invertida e incluso el aprendizaje basado en proyectos en los trabajos. Incluso sería interesante plantearse la realización de trabajos colaborativos, aunque exigiendo un gestor de versiones para poder ver y evaluar correctamente el trabajo de cada estudiante dentro del grupo. Con respecto a la gamificación en el aula, el uso de herramientas como Kahoot! a través de smartphones podría ayudar a hacer las clases más dinámicas.

## Referencias

- [1] Jesús R. Campaña, Ana E. Marín, María Ros, Daniel Sánchez, Juan M. Medina, M. Amparo Vila, M. Dolores Ruiz, Manuel P. Cuéllar y María J. Martín-Bautista. Metodologías activas y gamificación en las asignaturas de iniciación a la

	Sistemas Distribuidos	Ingeniería Informática Primer semestre		
	2017	2017	2016	2015
Intenta que el estudiante se interese por la asignatura	4,78	3,8	3,8	3,7
El profesor favorece la participación de los estudiantes en el desarrollo de la asignatura	4,26	3,8	3,7	3,7
El grado de conocimientos adquirido al cursar esta asignatura se corresponde con sus expectativas iniciales	4,22	3,8	3,8	3,8
Qué grado de interés tiene esta asignatura para su formación	4,57	3,7	3,6	3,9

Cuadro 3: Media de las respuestas obtenidas en las encuestas de evaluación y calidad del profesorado (Docencia-UR) en Sistemas Distribuidos durante el curso 2017 y en global en la Ingeniería Informática en los últimos 3 cursos. La escala es de tipo Likert de 5 puntos, siendo 1 lo peor valorado y 5 lo mejor valorado.

- programación. En *Actas de las XXII Jornadas de Enseñanza Universitaria de Informática, Jenui 2016*, páginas 245 – 252, Universidad de Almería, 2016.
- [2] Enrique A. de la Cal Marín, Marco A. García Tarmargo y María José Suárez-Cabal. Generación automática de pruebas de validación de prácticas de sistemas distribuidos. En *Actas de las XVI Jornadas de Enseñanza Universitaria de Informática, Jenui 2010*, páginas 523 – 526, Universidad de Santiago de Compostela, 2010.
- [3] Antoni Jaume-i-Capó, Isaac Lera, Francisco Juan Vives, Biel Moyà-Alcover y Carlos Guerrero Tomé. Experiencia piloto sobre el uso de la gamificación en estudios de grado de ingeniería en informática. En *Actas del Simposio-Taller previo a las XXII Jornadas de Enseñanza Universitaria de Informática, Jenui 2016*, páginas 35 – 40, Universidad de Almería, 2016.
- [4] Alberto Prieto Espinosa, Beatriz Prieto Campos y Begoña del Pino Prieto. Una Experiencia de flipped classroom. En *Actas de las XXII Jornadas de Enseñanza Universitaria de Informática, Jenui 2016*, páginas 237 – 244, Universidad de Almería, 2016.
- [5] Sílvia Terrasa Barrena y Gabriela Andreu García. Cambio a metodología de clase inversa en una asignatura obligatoria. En *Actas del Simposio-Taller previo a las XXI Jornadas de Enseñanza Universitaria de Informática, Jenui 2015*, páginas 32 – 37, Universitat Oberta La Salle, 2015.
- [6] Stephanie Weirich. How to give a good research talk. <https://www.cis.upenn.edu/~sweirich/talks/plmw15-giving-a-talk.pdf>, University of Pennsylvania, 2015.