

Animar al aprendizaje autónomo a través de gamificación en programación orientada a objetos

JORGE CALVILLO ARBIZU

Departamento de Ingeniería Telemática

Universidad de Sevilla

jcalvillo@us.es

D.O.I.: <http://dx.doi.org/10.12795/JDU.2018.i01.04>

Pp.: 80-95

Resumen

La programación orientada a objetos es una disciplina de difícil comprensión para los que se inician sin conocimientos previos y requiere de una ajustada combinación de teoría y práctica con el fin de adquirir las destrezas y habilidades necesarias para resolver problemas reales. Dentro de la enseñanza universitaria puede fácilmente perder su carácter práctico y convertirse en un conjunto de conceptos teóricos de difícil comprensión y aplicación en la práctica profesional.

Este trabajo presenta el diseño, desarrollo y evaluación de un ciclo de innovación docente en la asignatura Fundamentos de Programación II del Grado en Ingeniería de las Tecnologías de Telecomunicación de la Universidad de Sevilla en la que se pretende formar a los estudiantes en la programación orientada a objetos mediante el lenguaje Java. La metodología de innovación emplea elementos de gamificación y trabajos en pequeños grupos con el fin de motivar a los estudiantes y enseñar, a través de la autonomía del estudiante, tanto contenidos



Esta obra se distribuye con la licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional (CC BY-NC-ND 4.0.)

conceptuales como actitudes de resolución de problemas claves para los futuros ingenieros.

Palabras clave: Fundamentos de Programación II; Grado en Ingeniería de las Tecnologías de Telecomunicación, docencia universitaria; experimentación docente universitaria, gamificación.

Contexto

Fundamentos de Programación II es una asignatura obligatoria del primer curso del Grado en Ingeniería de las Tecnologías de Telecomunicación ofertado por la Escuela Superior de Ingeniería de la Universidad de Sevilla. La asignatura está dividida por igual en clases teóricas y prácticas. Las clases teóricas se imparten en un aula estándar con un volumen de alumnos entre 25 y 30. Las prácticas se realizan en el Centro de Cálculo de la Escuela Superior de Ingeniería con grupos de 12-15 alumnos.

Cabe destacar que, al tratarse de una asignatura de primer curso, los alumnos cuentan con limitados conocimientos en programación, y nulos cuando se trata de programación orientada a objetos, que es el foco de esta asignatura.

Diseño del ciclo de mejora docente

El ciclo de mejora docente se diseñó para los últimos temas de la asignatura centrados en Colecciones y Mapas, que no son más que estructuras de datos del lenguaje de programación Java.

Mapa de contenidos

En la Figura 1 se muestra el mapa de contenidos relacionados con estos dos temas. Los círculos azules tienen que ver con contenidos conceptuales y van desde lo más importante (azul oscuro) a lo menos importante (azul claro). Los conceptos Clase y Objetos ya forman parte de lo aprendido por los alumnos anteriormente en la asignatura

y a partir de ellos y la pregunta “¿Cómo agrupamos objetos?” podemos introducir los nuevos conceptos de Colecciones y Mapas.

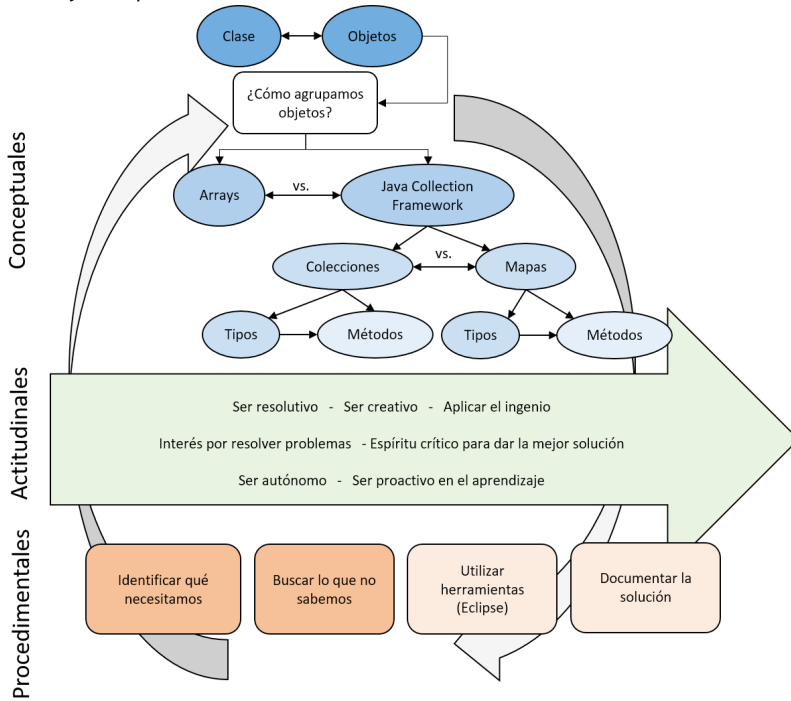


Figura 1. Mapa de contenidos para los temas Colecciones y Mapas de la asignatura Fundamentos de Programación II

Independientemente de los contenidos conceptuales concretos, en esta asignatura se busca que los alumnos adquieran una serie de contenidos actitudinales y procedimentales transversales. En cuanto a los primeros, se representan en la flecha verde y tienen que ver con la propia capacidad del alumno para resolver problemas de forma autónoma y tener espíritu crítico para dar la mejor solución, actitudes propias de cualquier ingeniero.

Los contenidos procedimentales son transversales, pero se particularizan para la asignatura concreta. Como más importantes están tanto la identificación de lo que el alumno desconoce como el conocimiento de las herramientas que nos permiten aprender lo que no sabemos. En programación existen multitud de fuentes de información

en Internet que pueden ayudar a cualquier a resolver dudas de manera autónoma.

Por otro lado, como contenidos procedimentales menos importantes tenemos la utilización de herramientas software para programar (no son necesarias, pero sí muy útiles) y la elaboración de documentación del código (esta actividad es muy ingrata y los alumnos son reacios a realizarla, aunque es una buena práctica indispensable en la labor del programador).

Las flechas añadidas al fondo indican que hay una fuerte interacción entre los contenidos conceptuales y los procedimentales (sobre todo en cuanto a la identificación y búsqueda de lo que el alumno desconoce) motivados siempre por actitudes como la proactividad o la creatividad.

Modelo metodológico y secuencia de actividades

El modelo metodológico diseñado cubría una sesión teórica en el aula tradicional y una sesión de práctica delante del ordenador, y estaba centrado en la resolución de problemas en pequeños grupos con un alto componente de investigación autónoma sin intervención docente. El objetivo es crear una experiencia de aprendizaje diversa donde se rompa con la metodología tradicional de aprendizaje y obligue a los estudiantes a “activarse” y ser participativos. Esta experiencia de aprendizaje se basa en la propuesta de Ken Bain (Bain, 2007) de crear un entorno de aprendizaje natural y crítico. El modelo metodológico posible adoptado se muestra en la Figura 2 y se rige por las etapas de problemas (P), hipótesis (H), búsqueda de información (BI) y solución (S).

Por un lado, para la sesión en el aula se escogió un modelo basado en gamificación que, a través de un juego de preguntas por grupos, exigía a los alumnos que expusieran sus conocimientos, investigaran sobre nuevos conceptos y tomaran decisiones basadas en el consenso

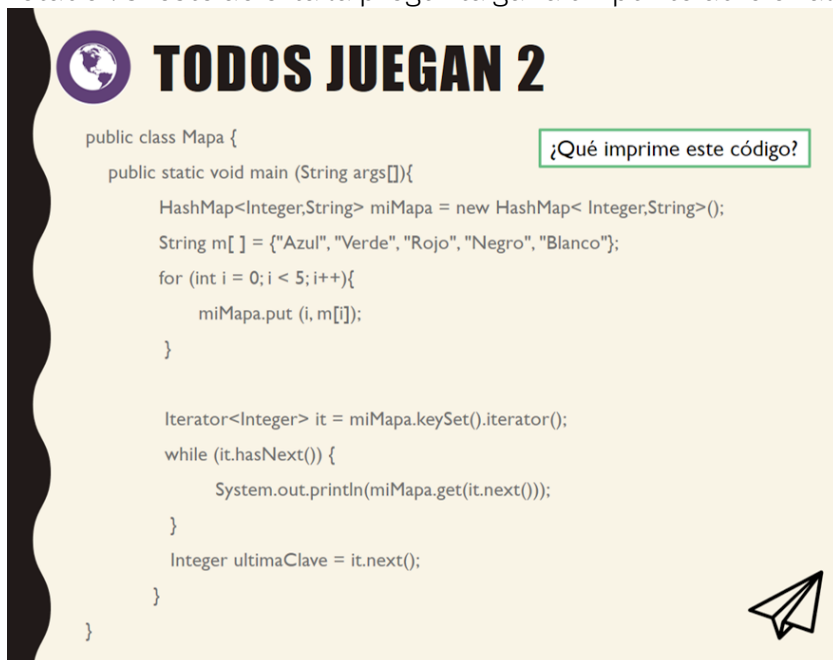
grupal. La sesión se planificó con una duración de 1 hora y 50 minutos y se formaron grupos de 4 alumnos. Cada grupo representaba un equipo que competía contra el resto en un juego de preguntas sobre los contenidos conceptuales específicos, es decir, sobre Colecciones y Mapas en Java. En cada ronda de juego, el equipo con menor puntuación elegía una pregunta que puede ser de los tipos: grupal, todos juegan y duelo (Figura 3).



Figura 2. Modelo metodológico posible

La pregunta grupal va dirigida al equipo que la escoge. Si la responde correctamente gana un punto. Si no, hay rebote y el siguiente equipo con menor puntuación tiene oportunidad de responder, y así sucesivamente. Las preguntas de tipo “todos juegan” tienen la particularidad de que todos los equipos responden a la pregunta y ganan puntos aquellos que la acierten. Finalmente, el equipo puede utilizar la pregunta duelo para retar a otro. En caso

de que el equipo retado falle, pierde un punto que gana el retador. Si este acierta la pregunta gana un punto adicional.



TODOS JUEGAN 2

```
public class Mapa {
    public static void main (String args[]){
        HashMap<Integer,String> miMapa = new HashMap< Integer,String>();
        String m[ ] = {"Azul", "Verde", "Rojo", "Negro", "Blanco"};
        for (int i = 0; i < 5; i++){
            miMapa.put (i, m[i]);
        }

        Iterator<Integer> it = miMapa.keySet().iterator();
        while (it.hasNext()) {
            System.out.println(miMapa.get(it.next()));
        }
        Integer ultimaClave = it.next();
    }
}
```

¿Qué imprime este código?

Figura 3. Conjunto de preguntas del juego y ejemplo de pregunta de tipo "todos juegan"

El juego de preguntas no era una prueba de conocimiento enmascarada como un juego, sino que las preguntas incluían conceptos que los alumnos no habían visto todavía. Por eso, para responderlas se daban varios minutos en los que ellos tenían que buscar la información que necesiten en los manuales, Internet, etc. El profesor era un simple moderador del juego y no respondía a ninguna cuestión conceptual.

Tras la resolución, los alumnos que hubieran acertado debían explicar siempre el razonamiento que les había llevado a la conclusión y los equipos que no habían acertado podían hacer preguntas para entenderlo. Esto obliga a los que aciertan a explicar a sus compañeros parte de la teoría. El papel del profesor es simplemente indicar la respuesta correcta y completar las explicaciones de los alumnos a sus compañeros. El juego termina cuando se

acaba la sesión y el desenlace es proclamar ganador al equipo con más puntos.

La segunda parte del ciclo de mejora se llevaba a cabo en la sesión práctica con lo que hemos llamado un taller de programación. En él los alumnos trabajan en parejas durante 1 hora y 50 minutos delante de un ordenador y siguiendo las instrucciones de una hoja de trabajo. Este modelo metodológico se basa en la propuesta de talleres conceptuales de Don Finkel (Finkel, 2008).

En este caso, tenían que desarrollar un programa a partir de una plantilla inicial añadiendo diferentes aspectos de Colecciones y Mapas que se cubren en el tema. Para ello e igual que pasaba en el juego de preguntas, debían ser proactivos en la búsqueda de información y la resolución de los problemas que se encontraran. Además, en la hoja de trabajo se incluían preguntas que obligaban a pararse a pensar y razonar sobre lo que se está haciendo.

(Extracto de la hoja de trabajo)

Mantenga abierto el fichero “Colecciones” que contiene la clase principal para analizar qué está ocurriendo en el ejemplo.

A continuación, ejecute *make*, que compilará y ejecutará el código ejemplo proporcionado. Si quiere disponer de la documentación en html también puede ejecutar *make -f makeJavadoc*.

Cuando la ejecución pare, esperando un *return/intro*, analice qué ha ocurrido y qué código se corresponde a ese comportamiento. Repita la operación cada vez que realice un ejercicio propuesto y haya modificado el código.

- Observe la declaración de la colección (Figura 4). *serieDeFiguras* será una referencia a una colección de tipo *ArrayList*, de elementos que serán implementaciones/realizaciones de la interfaz *Figura*. Observe

también el uso del método *isEmpty* para verificar que al crearse está vacía.

```
//Vamos a utilizar una colección de tipo ArrayList, una
implementación concreta de la interfaz List

List<Figura> serieDeFiguras = new ArrayList<Figura>();

    /*VERIFICAR QUE ESTÁ VACÍA: ISEMPY***/

System.out.println("**EJEMPLO: acabo de crear la coleccion, verifico
que esta vacia*\n");

if(serieDeFiguras.isEmpty())

    System.out.println("NO hay elementos en la coleccion");

else

    System.out.println("SI hay elementos en la coleccion");
```

Figura 4. Documentación de ejemplo de la hoja de trabajo

- A continuación, observe como crear figuras, añadir-las a la colección y verificar el tamaño. ¿Qué métodos se han utilizado? Piense ¿Qué diferencia práctica cree que habrá entre realizar el new previamente y luego usar la referencia al objeto en el add, como se hace con cuad1, 2 y 3, o incluir el new en el método de añadir, como se hace con el resto de figuras? ¿Cuándo usaría una u otra? Observe cómo puede imprimirse serieDeFiguras con el método println.

(Fin del extracto)

Cuestionario de exploración para el seguimiento de los estudiantes

A continuación se muestra el cuestionario utilizado antes y después de la experimentación. Las preguntas tienen un carácter eminentemente práctico, pero permiten valorar el conocimiento de los alumnos con respecto al tema que vamos a abordar.

Fundamentos de Programación II

Cuestionario Colecciones y Mapas

Nombre/Pseudónimo: _____

Fecha: _____

1. Si queremos implementar una agenda telefónica que guarde en orden un conjunto de personas y números de teléfono, ¿qué tipo de estructura de-bemos utilizar y por qué?
2. Para almacenar y gestionar un conjunto de objetos de una clase Ciudad, ¿qué estructura es más recomendable usar si no sabemos al programar cuántos objetos vamos a tener y por qué?
3. ¿Qué condición debe cumplir una clase (por ejemplo, clase Persona) si queremos tener una colección de objetos que pueda ordenarse?
4. ¿Con qué 2 métodos podemos recorrer una colección?

Aplicación del ciclo de mejora docente

El ciclo de mejora se aplicó, como estaba diseñado, en una sesión teórica y otra práctica. En la primera basada en el juego de preguntas los alumnos estuvieron altamente motivados durante toda la sesión, participativos y colaboradores dentro de los grupos. Parece que el componente de gamificación contribuyó a esta dosis extra de motivación.

Mantenerlos en grupos de 4 alumnos equilibró perfectamente la actividad entre el dinamismo para la búsqueda de información y la participación de todos los miembros. Cuando el tiempo de la sesión se acabó, los grupos que tenían menor puntuación aún querían nuevas oportunidades para remontar.

Como aspectos a mejorar podemos decir que algunas preguntas eran más difíciles que otras y exigían una búsqueda y razonamientos más complejos, y el orden en que aparecían no se había pensado previamente. La complejidad de las preguntas era aleatoria y eso hizo que algunas preguntas difíciles salieran al principio cuando los alumnos acababan de ponerse en contacto con los conceptos básicos. Habría que haber diseñado el juego de preguntas

para que la complejidad fuera in crescendo o haber puesto un apartado aparte de preguntas complejas con puntuaciones más altas, para empezar siempre por las básicas de introducción a los nuevos contenidos conceptuales.

La segunda parte del ciclo, el taller de programación transcurrió sin problemas y los alumnos resolvieron la mayor parte de las preguntas y ejercicios propuestos. Al reducir ahora los grupos a dos miembros el trabajo era más intenso, pero no tenían la sensación de estar solos y, además, tras el juego de preguntas ya estaban ligeramente familiarizados con los contenidos, no empezaban de cero. Por otro lado, el carácter más grupal y autónomo del taller permitió que el docente pudiese acercarse a los diferentes grupos y solucionar bloqueos cuando se quedaban atascados o dar alguna pista cuando no sabían responder a alguna pregunta.

Evaluación del aprendizaje de los estudiantes

En las Tablas 1-4 se resumen los resultados de los cuestionarios de exploración. A partir de las respuestas a cada pregunta en la exploración PRE se han identificado una serie de modelos conceptuales que van desde un conocimiento nulo a un conocimiento completo (Figuras 5-8). En el caso concreto de la pregunta 2 se identificó un modelo adicional en la exploración POST.

Cabe señalar que la exploración PRE la realizaron 9 alumnos y la POST 13. Pese a ello, los 13 participaron en el segundo ciclo de mejora, aunque no asistieron a la clase previa donde se realizó la exploración PRE.

Tabla 1
Escalera de aprendizaje y resultados Pre y Post de la pregunta 1

Pregunta 1. Si queremos implementar una agenda telefónica que guarde en orden un conjunto de personas y números de teléfono, ¿qué tipo de estructura debemos utilizar y por qué?



PRE (9 alumnos)	Modelos	POST (13 alumnos)
2	A) No lo sé	3
3	B) Respuesta al azar sin razonamiento	2
4	C) Respuesta correcta, pero razonamiento incompleto	5
0	D) Respuesta correcta y razonamiento completo	3

Figura 5. Escalera de aprendizaje de la pregunta 1

Tabla 2
Escalera de aprendizaje y resultados Pre y Post de la pregunta 2

Pregunta 2. Para almacenar y gestionar un conjunto de objetos de una clase Ciudad, ¿qué estructura es más recomendable usar si no sabemos al programar cuántos objetos vamos a tener y por qué?		
PRE (9 alumnos)	Modelos	POST (13 alumnos)
3	A) No lo sé	3
1	B) Respuesta al azar sin razonamiento	3
3	C) Respuesta correcta, pero razonamiento incompleto	3
2	D) Respuesta correcta y razonamiento completo	3
	E) Respuesta correcta, razonamiento completo y contraste con otras opciones	1

Figura 6. Escalera de aprendizaje de la pregunta 2

Tabla 3
Escalera de aprendizaje y resultados Pre y Post de la pregunta 3

Pregunta 3. ¿Qué condición debe cumplir una clase (por ejemplo, clase Persona) si queremos tener una colección de objetos que pueda ordenarse?		
PRE (9 alumnos)	Modelos	POST (13 alumnos)
6	A) No lo sé	3
3	B) Respuesta al azar sin razonamiento	4
0	C) Respuesta correcta, pero razonamiento incompleto	2

0	D) Respuesta correcta y razonamiento completo	4
---	---	---

Figura 7. Escalera de aprendizaje de la pregunta 3

Tabla 4
Escalera de aprendizaje y resultados Pre y Post de la pregunta 4

Pregunta 4. ¿Con qué 2 métodos podemos recorrer una colección?		
PRE (9 alumnos)	Modelos	POST (13 alumnos)
2	A) No lo sé	0
1	B) Respuesta incorrecta	2
2	C) Un método bien	2
2	D) Un método bien y el otro mal	1
2	E) Los dos métodos bien	8

Figura 8. Escalera de aprendizaje de la pregunta 4

Los resultados muestran en líneas generales un avance del conocimiento de los alumnos tras la aplicación del segundo ciclo de mejora. Igualmente aparecen, aunque en menor medida, alumnos que se mantienen en los escalones más bajos en la exploración POST.

En la Tabla 5 se muestran los resultados individuales para los 9 sujetos que hicieron los dos cuestionarios.

Tabla 5
Resultados diferenciales de los estudiantes para las 4 preguntas del cuestionario

	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4
Sujeto 1	+1	=	+1	+1
Sujeto 2	+1	=	=	+3
Sujeto 3	+1	+1	+2	=
Sujeto 4	+1	+1	+2	=
Sujeto 5	+1	+2	=	+3
Sujeto 6	=	=	+4	+2
Sujeto 7	=	+3	+2	=
Sujeto 8	+1	=	+2	=
Sujeto 9	=	=	+2	+4

Evaluación del ciclo de mejora docente

En relación con la metodología docente y acorde con lo expuesto en (Porlán, 2017), uno de los puntos clave que se pueden extraer del ciclo de mejora aplicado es que en el aprendizaje no solo cuenta el contenido a impartir y nosotros como docentes. Los alumnos son una pieza fundamental que debemos conocer para adaptar así nuestra enseñanza y para poder evaluar el proceso de aprendizaje en todo su desarrollo (y no simplemente al final cuando se evalúa la asignatura).

Por otro lado, los contenidos merecen tanto una reflexión inicial como continuada a lo largo del proceso de aprendizaje. A menudo nos quejamos del “poco tiempo” que tenemos para dar unos contenidos que nos imponen o nos imponemos y no nos paramos a pensar en qué contenidos son fundamentales y cuáles no. Esta reflexión debemos hacerla antes de diseñar el proceso de aprendizaje y tenerla presente durante todo el curso, ajustando la metodología en función de los estudiantes y esa priorización de los contenidos. También es importante diferenciar los contenidos entre aquellos que los alumnos pueden adquirir por su cuenta y los que requieren de una asistencia docente. Este punto también permite priorizar unos contenidos sobre otros, pero siempre que seamos “valientes” y rompamos la barrera psicológica de dejar contenidos sin explicar y que el alumno los estudie autónomamente (aunque sea con documentación preparada por el docente). Para estas reflexiones sobre los contenidos, los mapas conceptuales se han mostrado como una herramienta muy interesante para desarrollar una visión general no solo de los contenidos conceptuales sino también de los procedimentales y actitudinales.

Una vez más, la reflexión antes, durante y después del proceso de aprendizaje es una herramienta que deberíamos poner en práctica todos los docentes. El proceso de aprendizaje debe estar descompuesto en actividades,



descritas y programadas en el tiempo a modo de guion que nos ayude a no desviarnos. En cuanto a las actividades, la utilización de preguntas-claves relacionadas con la realidad de los alumnos potencia la motivación de estos y les ayuda también a incluir los nuevos contenidos conceptuales en sus mapas conceptuales.

Para ayudarnos con la reflexión de nuestra práctica docente es interesante la propuesta consistente en llevar un diario con el desarrollo de las actividades, las impresiones que extraemos de los alumnos, sus dudas y las dificultades que encontramos para alcanzar nuestro modelo metodológico deseado (Porlán, 2017). La exploración del conocimiento de los alumnos con cuestionarios pre-post supone un mecanismo interesante no solo para comprobar cómo de desarrollados están los modelos conceptuales de los alumnos (y actuar en consecuencia para ayudarles a avanzar en la “escalera del aprendizaje”) sino también para comprobar la efectividad del propio proceso de aprendizaje.

Principios didácticos susceptibles de ser incorporados a la práctica habitual

La experiencia con los ciclos de mejora y el aprendizaje realizado en el curso nos anima a incorporar a la práctica docente los siguientes principios didácticos:

- Analizar a priori los conocimientos de los estudiantes y partir de ahí para desarrollar el proceso de aprendizaje.
- No presentar los contenidos conceptuales sin un razonamiento previo de los estudiantes a través de la resolución de problemas concretos que los motiven.
- Preparar las sesiones a través de la elaboración de mapas de contenidos y secuencias de actividades.

- Potenciar el razonamiento crítico de los estudiantes y fomentar su inquietud por descubrir de manera autónoma.
- Utilizar diario del profesor para hacer un seguimiento del desarrollo del aprendizaje y poder reflexionar sobre el mismo y hacer modificaciones si es necesario.
- Diversificar el tipo de actividades y analizar cuáles tienen mayor impacto en el aprendizaje de los estudiantes.
- Proporcionar materiales complementarios o referencias para los estudiantes más adelantados.
- Incluir actividades de autoevaluación de los estudiantes.

Referencias bibliográficas

- Bain, K. (2007). *Lo que hacen los mejores profesores universitarios*. Valencia: Universitat de Valencia. Servei de Publicacions.
- Finkel, D. (2008). *Dar clase con la boca cerrada*. Valencia: Universitat de Valencia. Servei de Publicacions.
- Porlán, R. (2017). *Enseñanza universitaria. Cómo mejorarla*. Madrid: Ediciones Morata.

