

Mejora de la docencia de la programación usando placas IoT micro:bit y robots GiggleBot con metodologías PBL y aprendizaje lúdico en una asignatura de primer curso de grado

BANCO DE BUENAS PRÁCTICAS DOCENTES

JOSÉ CENTENO GONZÁLEZ

PEDRO DE LAS HERAS QUIRÓS

 Universidad
Rey Juan Carlos



CENTRO DE INNOVACIÓN EN EDUCACIÓN DIGITAL
Universidad Rey Juan Carlos

I. La práctica

- **Título:** Mejora de la docencia de la programación usando placas IoT micro:bit y robots GiggieBot con metodologías PBL y aprendizaje lúdico en una asignatura de primer curso de grado.
- **Curso Académico:** 2019/2020
- **Asignatura:** Fundamentos de la programación
- **Área/Titulación:** Ingeniería de Robótica Software
- **Grupo de Estudiantes:** Grupo completo de 1º del Grado
- **Palabras clave:** fundamentos de programación; CS1

El objetivo de la práctica docente aquí descrita fue conseguir, mediante cambios en la metodología docente, que el estudiante de Ingeniería de Robótica Software de primer curso adquiriese un aprendizaje más robusto basado en la interacción con hardware real al que debe dotar de un comportamiento requerido en la especificación de las tareas.

Para ello se llevaron a cabo cambios metodológicos incorporando las prácticas docentes de la comunidad DBER [5] consistentes en el fomento de la parte práctica de la asignatura asignando tareas como medio de aprendizaje para las que los profesores proporcionan realimentación detallada en tiempo real, y el fomento del aprendizaje social mediante el diseño de prácticas en grupo que incorporan aspectos de aprendizaje lúdico (gamification) y de aprendizaje cooperativo.

Como soporte a la docencia de la práctica docente descrita se ha utilizado un libro multimedia interactivo con simuladores de código y pruebas de evaluación formativa integradas. Gracias a esta herramienta digital, los profesores superamos el ritmo tradicional de “exposición magistral” de los conceptos de la asignatura.

Para reforzar el aprendizaje práctico se ha utilizado hardware específico para la asignatura, que se ha entregado a los estudiantes. Todas estas herramientas y tecnologías digitales se han gestionado desde el encomiable entorno del Aula Virtual de la URJC.

La práctica docente se planteó a los estudiantes para su desarrollo a lo largo de aproximadamente dos tercios de la duración del cuatrimestre, quedando el primer tercio para la explicación de los aspectos más básicos del lenguaje puestos en práctica también mediante aprendizaje activo basado en el módulo turtle de Python.

Una de las metodologías docentes en las que se fundamenta esta propuesta es PBL (*Problem Based Learning*), mediante el establecimiento de diversas fases hasta completar el proyecto de programación, cada una de ellas planteada con sus propios objetivos que el estudiante debe alcanzar.

Además, se han incluido aspectos de aprendizaje lúdico en el proyecto a desarrollar, planteando unas pruebas competitivas en un entorno físico no predecible, y premiando a los estudiantes cuyo robots, gracias a la estrategia por ellos diseñada y programada, alcanzaran antes el cumplimiento de los retos planteados.

Por último, una última fase del proyecto se basaba en *Peer Instruction*, debiendo los estudiantes formar equipos cuyos robots cooperasen para cumplir antes los retos planteados.

Se han diseñado pruebas de evaluación de distinto tipo para cada etapa de la práctica, cuyas rúbricas de evaluación están diseñadas para reforzar el cumplimiento de los objetivos de esta práctica docente.

Se realizó un análisis cuantitativo de los resultados académicos de los estudiantes tanto en esta asignatura como en la que la sucede durante 2 cursos académicos, para poder así comparar los resultados de aprendizaje. Los resultados de las evaluaciones sumativas han sido muy similares en cuanto a números de estudiantes calificados con Sobresaliente, Notable o Aprobado. Sin embargo, gracias a esta práctica docente hemos conseguido ejercer en el alumnado un efecto motivador, habiéndose duplicado el porcentaje de estudiantes que entregan la última etapa del proyecto final, y el porcentaje de estudiantes que reciben una calificación en el primer cuartil en la evaluación sumativa relativa a su proyecto.

La mejora de los resultados en la asignatura que sucede a esta ha sido muy significativa, duplicándose en porcentaje los notables y los aprobados, y habiéndose reducido a la mitad los suspensos, sin que se hayan realizado cambios metodológicos o de otra índole en esta otra asignatura. De esta forma consideramos cumplido el objetivo fundamental de los cambios en nuestra asignatura.

Por último, se realizó una encuesta basada en el *Student Evaluation of Educational Quality Questionnaire (SEEQ)*, cuyos resultados fueron que los estudiantes aprecian de manera muy destacada esta práctica docente frente a otras más convencionales, y lo consideran muy mayoritariamente una parte “muy valiosa” de la asignatura.

Puede consultarse documentación adicional (enunciados de las etapas de las prácticas, fotos y vídeos de la competición/evaluación del proyecto final, y encuestas sobre la asignatura) en el siguiente [enlace](#).

2. Justificación

La primera asignatura de Programación en el primer cuatrimestre de un grado en el ámbito de las TIC, conocida como CS1 en el ámbito académico estadounidense, resulta permanente objeto de estudio en las universidades del mundo entero, y existen múltiples enfoques y paradigmas para ella [1]. Aún teniendo en consideración todos esos estudios, es importante evaluar el contexto de cada asignatura concreta en un grado en particular y con un perfil

de ingreso de estudiantes específico antes de decidir utilizar una u otra metodología docente para mejorar los resultados académicos de una asignatura.

En el grado de Ingeniería de Robótica Software varios estudiantes llegan a su primera asignatura de programación con cierta experiencia previa utilizando distintos lenguajes de programación como C, C++, Java, Python, Javascript, Scratch o Processing (según encuesta realizada el primer día a los alumnos de la asignatura). Sin embargo en un gran número de casos se aprecian serias deficiencias en las habilidades básicas de programación de los alumnos de nuevo ingreso ya que mucha de su formación anterior se limita a transcribir o adaptar levemente ejemplos de código que se entregan a priori al estudiante, o a programar pequeños algoritmos finalistas para conseguir que robots utilizados en enseñanzas medias realicen algún tipo de movimiento o acción, no siendo el objetivo de este tipo de asignaturas proporcionar una enseñanza sólida de la programación de ordenadores. En encuestas propias realizadas a los estudiantes el primer día de clase de la asignatura en la que se ha aplicado la práctica docente descrita en esta memoria un 75% del grupo revelaba saber programar “solo un poco”. Además, un porcentaje significativo del grupo, el 15%, manifestaba no saber nada de programación previamente a la asignatura. Esto supone que el 90% de los estudiantes tienen un reto significativo por delante para alcanzar los resultados de aprendizaje previstos en esta asignatura de primer cuatrimestre, de forma que puedan abordar con confianza muchas otras asignaturas de su grado, algunas de ellas en el segundo cuatrimestre del mismo primer curso.

Muchos de los debates académicos sobre el planteamiento de una asignatura CS1 se centran en elegir el lenguaje de programación más adecuado para los objetivos académicos perseguidos. Sin embargo, los profesores de esta asignatura estamos convencidos que es mucho más importante elegir con detenimiento las metodologías docentes más apropiadas, que seleccionar un lenguaje de programación concreto. Por esta razón hemos aceptado el lenguaje recomendado por la Coordinación del grado (Python3, un lenguaje “habitual” en el mundo de la programación de robots) pese a que probablemente no hubiera sido nuestra primera elección, ni lo será en el futuro. Cuando hace unos años gran parte de la comunidad académica entendió que había que adoptar, precisamente, el lenguaje Python en CS1, porque así lo había hecho el MIT siguiendo las recomendaciones del profesor Hal Abelson, no todos repararon en el cambio de metodología y de plataforma de prácticas que el MIT empleó para conseguir que sus estudiantes aprendiesen a programar resolviendo problemas complejos, independientemente del lenguaje de programación elegido. Menos aún leyeron la justificación que el profesor Hal Abelson dio para usar Python. Nosotros sí lo hicimos antes de aceptar el reto de utilizar un lenguaje de programación tan poco idóneo para enseñar a programar. Desgraciadamente cientos de universidades en todo el mundo creyeron que era el lenguaje, y no las metodologías ni la plataforma de prácticas lo que había que cambiar.

Nuestro planteamiento para la asignatura sin embargo, teniendo en cuenta el análisis detenido de la experiencia que realizó el profesor Hal Abelson, fue hacerla eminentemente práctica, cambiando la metodología y la plataforma de prácticas. Todas las clases se imparten en un laboratorio con ordenadores que utilizan el sistema operativo Ubuntu GNU/Linux con conexión a Internet, y las explicaciones de los conceptos del lenguaje se van entrelazando con continuos ejercicios que son utilizados para realizar evaluaciones formativas con realimentación inmediata, que el estudiante va realizando dentro y fuera de las clases en el libro interactivo multimedia que se utiliza en la asignatura [6]. Para los proyectos de programación de más tamaño necesitábamos una plataforma que consiguiera desplegar las metodologías docentes que permitieran motivar al estudiante en su aprendizaje y mejorar los resultados de aprendizaje.

En el curso 2018–2019, el primero en que se impartió esta asignatura, los profesores elegimos que los estudiantes desarrollaran parte de sus proyectos de programación utilizando la plataforma hardware *micro:bit* diseñada para programar dispositivos para la Internet de las cosas (*Internet of Things—IoT*) que pueden ser utilizados autónomamente para recibir señales a través de sus propios sensores o de otros conectados al *micro:bit*, para controlar robots, etc.

Para estos proyectos basados en *micro:bit* decidimos utilizar prácticas docentes que habitualmente se relacionan con las metodologías PBL [3], *Peer Instruction* [4] y *Gamification* [2]. Sin embargo, como se describe en la sección sobre aspectos metodológicos de esta memoria, no se adoptó una metodología, sino que se realizó un análisis de las mejores prácticas docentes utilizando el marco metodológico de la comunidad de investigadores sobre la docencia universitaria DBER [5].

En estas prácticas se planteaban al estudiante las especificaciones del proyecto a realizar, utilizando las clases en el laboratorio para la monitorización por parte de los profesores del trabajo en curso y para la resolución de problemas, y se proporcionaba a los estudiantes una realimentación precisa y en el momento.

Esta práctica docente se complementó con el fomento del aprendizaje social, para lo cuál se le planteó a los estudiantes un reto opcional con aspectos de aprendizaje lúdico en el que los estudiantes debían competir entre sí utilizando los *micro:bit* al aire libre en el arboretum del campus de Fuenlabrada, ejecutando un programa que debían desarrollar los alumnos para completar unos retos antes que sus compañeros consistentes en descubrir *micro:bit* ocultos entre el follaje del arboretum mediante la medición de la potencia de la señal de radio recibida. Los alumnos conseguían con ello una mejora en la calificación y unos premios para los 3 mejores resultados consistentes en 3 libros de algoritmos y estructuras de datos.

Tras la experiencia de ese primer curso académico, en este curso 2019-2020 hemos ampliado y extendido esta práctica docente, conformando la que presentamos en la presente propuesta, y que incluye:

- Utilización de *micro:bits*. Se amplió el número de *micro:bits* entregados en préstamo durante el curso a cada estudiante, pasando de 1 a 2.
- Utilización de *GiggleBots*. El *GiggleBot* es un robot diseñado y fabricado por Dexter Industries, compañía estadounidense que lleva 10 años trabajando en robots para entornos educativos, siendo éste su primer robot para la plataforma *micro:bit*. Se entregó a cada estudiante 1 *GiggleBot* en préstamo durante el curso.
- Desarrollo del proyecto final de la asignatura totalmente basado en *micro:bit* + *GiggleBot* utilizando la metodología PBL. El proyecto consistía en realizar un programa de ordenador en el lenguaje Python que ejecutado en el *micro:bit* insertado a bordo del *GiggleBot* debía localizar mediante señales de radiofrecuencia la presencia de otros *micro:bit* balizas distribuidos en emplazamientos elegidos al azar en un laboratorio, y tras obtener de cada uno ellos un mensaje secreto, entregar dichos mensajes a un *micro:bit* árbitro situado en una zona iluminada que los robots debían descubrir mientras deambulaban evitando obstáculos utilizando un algoritmo de evitación de obstáculos.
- Búsqueda de balizas por parte del robot, para lo cuál los estudiantes debían concebir estrategias de búsqueda y programarlas en los robots.
- Extensión de los aspectos de aprendizaje lúdico a todo el proyecto final, aplicándose a todos los estudiantes, de forma que los programas desarrollados por cada estudiante fueron evaluados en un “campo de pruebas” desplegado en un laboratorio dedicado en exclusiva a esta práctica durante unas semanas (laboratorio del que pudieron disponer los estudiantes antes de la prueba de evaluación, aunque sin conocer los emplazamientos finales de las balizas ni el árbitro), compitiendo entre ellos por alcanzar los objetivos en el menor tiempo posible.
- Complemento de aprendizaje lúdico con *Peer Instruction* en la última etapa (opcional) del proyecto, debiendo formar equipos de hasta 3 estudiantes que cooperasen en la detección de balizas repartiendo la tarea entre sus robots para poder completar la prueba en un menor tiempo gracias al trabajo en equipo. Para ello los estudiantes tuvieron que colaborar en la generación de nuevas estrategias de trabajo colaborativo entre los robots. En la siguiente figura se puede ver una imagen de uno de los equipos el día de la competición y evaluación de las prácticas:



En el sitio web referenciado en el resumen se puede consultar el enunciado completo de las distintas fases del proyecto final, así como más fotos y vídeos de los ensayos de los estudiantes y de la prueba de competición/evaluación final.

3. Desarrollo

Objetivos

El objetivo de esta práctica docente es conseguir, mediante cambios en la metodología docente, que el estudiante adquiriera un aprendizaje más robusto basado en la interacción con hardware real al que debe dotar de un comportamiento requerido en la especificación de las tareas. De manera más exhaustiva, se detallan a continuación los objetivos tanto generales como específicos.

Objetivos generales

1. Aumentar el número de horas dedicado por el estudiante a la realización de actividades de índole práctica frente al estudio teórico de la asignatura.
2. Aumentar la motivación y el interés del estudiante por el campo de la programación, mediante la interacción con el hardware que le es propio al ámbito de su grado de Ingeniería de Robótica Software.
3. Incrementar las capacidades del estudiante en el análisis y toma de decisiones propias y diferenciadas de las de otros estudiantes mediante el incentivo de que sus soluciones resulten las más eficaces en la competición final usando técnicas de aprendizaje lúdico.
4. Acostumbrar al estudiante a la utilización de materiales y tecnologías abiertas: todos los materiales de la asignatura incluyendo el libro electrónico, la documentación elaborada por los profesores y la propia plataforma hardware micro:bit son abiertos (se publican con licencias tipo Creative Commons) e

incentivan que el estudiante busque más documentación y otros materiales abiertos que puedan ser de aplicación a sus proyectos.

5. Utilizando el aprendizaje social, mejorar el proceso de aprendizaje y desarrollar en el estudiante la capacidad de trabajar en equipo. Para ello se diseñó una última fase del proyecto en la que los estudiantes necesariamente tuvieron que concebir colaborativamente soluciones, y plasmarlas en el software que programaron para los robots, debiendo desarrollar dichas soluciones entre varios estudiantes de manera cooperativa, obteniendo si lo conseguían una mejora adicional en su calificación.

Objetivos específicos

1. Reforzar el aprendizaje del ciclo de desarrollo de software utilizando metodologías ágiles para capturar requisitos, especificar, diseñar, programar, probar y desplegar el software desarrollado utilizando una plataforma hardware que muestra de manera plástica al estudiante el resultado de las acciones del software desarrollado sobre los sensores y actuadores del robot, y sus posibles defectos y aspectos susceptibles de mejora.
2. Introducir en el contexto de una asignatura básica de programación de primer cuatrimestre de primer curso la interacción con sensores y actuadores de un robot que resultará útil para varias asignaturas del grado, en lugar de utilizar entornos alejados de la práctica profesional.
3. Introducir en el contexto de una asignatura básica de programación la comunicación por radio entre dispositivos y entre robots, que resultará útil para varias asignaturas posteriores del grado.
4. Fomentar en el estudiante la programación de forma robusta al someter a sus proyectos a la interacción con un entorno de pruebas real y cambiante, de forma que el estudiante intente predecir los posibles problemas de su diseño e implementación, y anticiparse a dichos problemas en la fase de desarrollo. Este objetivo es importante incluso en asignaturas de primer curso en el ámbito universitario, un aspecto que muchas veces no es tenido en cuenta al diseñar asignaturas de primer curso dejando para más adelante la consecución de este objetivo.

Metodologías docentes aplicadas

En primer lugar se describe el marco metodológico utilizado en la asignatura en la que se enmarca la práctica educativa, describiendo brevemente las metodologías utilizadas en la asignatura. Después se detalla la metodología utilizada en la práctica educativa innovadora descrita en esta memoria y finalmente, a modo de conclusión, se aportan unas conclusiones metodológicas.

Marco metodológico

Lejos quedan ya los tiempos en los que todo profesor consideraba y defendía que era un experto no sólo en su disciplina, sino también en la didáctica de ésta. En las últimas décadas los profesores se han vuelto más humildes y se han concebido en el ámbito académico de la enseñanza universitaria múltiples metodologías docentes, siendo la mayoría del profesorado más honesto y consciente de que debe reevaluar su metodología docente. Esta concienciación colectiva ha hecho que el mundo universitario de todo el planeta esté inmerso en un proceso histórico que es de esperar que nos permita transitar desde un modo de funcionamiento artesanal precientífico en lo referente a la docencia hasta un modo de funcionamiento basado en el método científico.

Como resultado de este proceso de mejora de la docencia instanciado en las múltiples metodologías docentes propuestas en todo el mundo, se han generado una serie de prácticas docentes que de manera objetiva y medible, cualitativa y cuantitativamente, permiten que los estudiantes alcancen unos mejores resultados que cuando se utilizan los métodos de enseñanza tradicionales heredados de otros siglos.

Desafortunadamente, a pesar de las muchas metodologías docentes que han aparecido, no toda la investigación realizada en este ámbito de la docencia universitaria es aprovechable por tres razones principales:

- Algunas de las metodologías están diseñadas para ámbitos muy específicos de la docencia universitaria, no resultando sus prácticas docentes aplicables a todas las disciplinas académicas, algo que frecuentemente provoca fracasos y desesperación en el profesorado y en el alumnado que es sometido a metodologías pensadas para disciplinas académicas muy distintas e incompatibles.
- Algunas metodologías no han dado lugar a estándares consensuados que puedan ser utilizados para evaluar la calidad del trabajo académico de un modo objetivo cuantitativo, por lo que la adopción de estas metodologías, por muy en boga que estén y por mucho que flipen a los más entusiastas, no es evaluable empíricamente incluso cuando se pretende dicho objetivo. De lo que no se puede medir no se pueden derivar conclusiones acerca de su calidad. Los buenos profesores son frecuentemente descritos en base a características personales como el entusiasmo, la cercanía con los estudiantes o el interés por su asignatura, elementos todos ellos deseables pero poco objetivables y no relacionables con los resultados de aprendizaje.
- Incluso cuando una metodología ha sido diseñada para asignaturas de la disciplina académica a la que se pretende aplicar, y cuando ha dado lugar a estándares consensuados, la aplicación holística a todas las actividades de una asignatura puede conllevar resultados no óptimos pues, especialmente en disciplinas académicas del ámbito de las ciencias y la ingeniería, es habitual que en una misma asignatura se desarrollen

a lo largo del curso académico actividades docentes que requieren utilizar una combinación adecuada de varias metodologías, y no una sola.

Estas razones han hecho que en muchos casos, a pesar de la concienciación del profesorado universitario de que hay que adoptar nuevas metodologías que mejoren los resultados de aprendizaje, muchos profesores hayan quedado defraudados cuando tras adoptar con entusiasmo una metodología novedosa no han observado una mejora de los resultados, atribuible a, entre otras, las tres razones antes mencionadas, independientemente de las notas que finalmente hayan decidido aplicar a sus estudiantes.

Los profesores que han llevado a cabo la práctica educativa descrita en esta memoria eran conocedores de los problemas antes descritos por haber sufrido varios de ellos o haber observado como compañeros caían en alguno o varios de dichos problemas. Ello los llevó hace varios años a abordar el estudio exhaustivo de la literatura científica para poder paliar estos problemas en la docencia de sus asignaturas.

En primer lugar, excluyeron los trabajos de investigación llevados a cabo en el ámbito de la enseñanza preuniversitaria. En segundo lugar, centraron su atención en los trabajos que abordaban la enseñanza universitaria, pero centrada en la ciencia y especialmente en la ingeniería, lo que les permitió encontrar prácticas docentes centradas en enseñar a los alumnos a pensar y a utilizar los resultados de aprendizaje de sus asignaturas para comportarse como un profesional de su disciplina. La búsqueda de prácticas docentes aplicables se centró en aquellas que permitiesen que los alumnos, al acabar la asignatura, pudiesen tomar decisiones relevantes razonando como lo harían los expertos en la disciplina de la asignatura.

Como resultado de este proceso de búsqueda se llegó a la conclusión de que independientemente de las metodologías adoptadas, las prácticas docentes empleadas en las asignaturas deberían basarse en la potenciación de la práctica en todas las actividades docentes, pues es la práctica la que permite enseñar a los alumnos a comportarse como los expertos de la disciplina en los ámbitos de la ciencia y la ingeniería. Cuanta más duración tengan las prácticas y más prácticas haya, mayor será el aprendizaje en estas disciplinas académicas, algo que bien entrado el siglo XXI aún hay que explicar a muchos compañeros de profesión, con poco éxito habitualmente, posiblemente debido al mayor nivel de esfuerzo requerido por parte del profesorado para llevar a cabo estos cambios en la docencia universitaria.

Se llegó así a la conclusión de que la enseñanza es efectiva cuando se llevan a cabo estas dos acciones:

- El diseño de actividades de aprendizaje que hacen que el estudiante tenga que realizar tareas prácticas en las que tiene que tomar decisiones, usando para ello procesos de razonamiento que incluyen el aprendizaje de conocimiento de manera autónoma.

- La provisión por parte del profesorado de realimentación al alumno a tiempo, específica, que no le desmotive, y que sea utilizable para mejorar por parte del alumno. Para ello los profesores diseñaron mecanismos de observación del desempeño de los alumnos y de comunicación de los resultados de dicha observación que se plasmaron en las pruebas de evaluación formativa descritas en esta memoria.

Dentro del campo de la investigación sobre la docencia universitaria cabe destacar el trabajo de la comunidad DBER (Discipline-Based Education Research), que en las últimas décadas ha desarrollado una labor exhaustiva de objetivación de esta disciplina académica [5]. Su foco está en la enseñanza universitaria de grado en las áreas de la ciencia, la tecnología, la ingeniería y las matemáticas, razón por la cuál sus resultados se aplican a esta asignatura.

El marco metodológico de esta comunidad no propone una metodología concreta, sino que proporciona mecanismos que pueden ayudar a implementar una enseñanza universitaria medible, basada en evidencias científicas, que está de acuerdo con el principio de la enseñanza basada en la práctica llevada a cabo por el estudiante combinada con la realimentación del profesor.

Metodologías utilizadas en la práctica educativa

En diferentes partes de la asignatura se han aplicado aspectos metodológicos que aparecen reflejados en prácticas docentes integradas en las siguientes metodologías: *Peer Instruction* [4], *Project Based Learning* [3], *Gamification* [2]. Sin embargo, ninguna de estas metodologías fue adoptada en su conjunto, ni siquiera en partes de la asignatura. Se adoptó tal como se ha explicado en la sección anterior una estrategia de abajo arriba, en la que se definieron prácticas docentes específicas para las cuáles la comunidad DBER ha constatado empíricamente que producen mejoras de resultados medibles.

Es habitual que en la evaluación de los aspectos metodológicos de una asignatura los profesores definan el esfuerzo metodológico llevado a cabo como una variante de una o varias metodologías, e incluso que lo califiquen como una nueva metodología.

Creemos que es más productivo definir las prácticas docentes concretas utilizadas y dejar a otros la labor de definir nuevas metodologías o nuevos nombres de nuevas metodologías.

Dentro del ámbito de la enseñanza universitaria de grado en ciencias, tecnología, ingeniería y matemáticas, y siguiendo el principio general de maximizar la práctica y dar realimentación al alumno, la comunidad DBER define un conjunto de principios específicos y de prácticas docentes que pueden conducir de manera medible a un mejor aprendizaje basado en la investigación docente. No describiremos aquí los principios, centrándonos en las dos

siguientes secciones en el análisis de cómo se han aplicado las dos prácticas docentes DBER a la práctica educativa descrita en esta memoria.

A. Tareas y preguntas con materiales entregables

En cuanto a las prácticas docentes, en primer lugar para que los estudiantes practiquen el tipo de pensamiento que los expertos de la disciplina llevan a cabo, hay que ofrecer a los estudiantes tareas y preguntas que requieran explícitamente este pensamiento. Los estudiantes deben proporcionar como resultado de la realización de estas tareas materiales entregables que permitan al profesor dar una realimentación positiva.

Los enunciados de la práctica educativa proporcionados a los alumnos que se pueden consultar en el sitio web referenciado en el Resumen permiten observar que están diseñados precisamente con este objetivo.

En cuanto a los materiales proporcionados por los alumnos como evidencia de la realización de las tareas, estos consistieron en:

- La entrega de programas desarrollados para los robots *gigglebot*
- La ejecución de éstos en tiempo real delante de los profesores
- La explicación oral de la ejecución de dichos programas en tiempo real por parte de los alumnos

La evaluación formativa y sumativa de estas evidencias de aprendizaje proporcionada por los alumnos se llevó a cabo mediante las rúbricas de evaluación descritas en esta memoria.

Si bien esta memoria se circunscribe a una práctica educativa concreta, es importante indicar que en el resto de actividades formativas de la asignatura se emplearon estos mismos principios DBER para realizar evaluaciones formativas en otras etapas de la asignatura con el objetivo de proporcionar realimentación a lo largo de todo el aprendizaje. Un ejemplo son las realimentaciones mediante la corrección en el mismo día de los ejercicios integrados en el libro interactivo multimedia descrito en esta memoria, o las preguntas rápidas lanzadas por el profesor que son respondidas por los alumnos y corregidas por el profesor durante las clases en las que el profesor combina las explicaciones cortas con la realización de ejercicios en el laboratorio por parte de los alumnos.

B. Aprendizaje social

La interacción entre los alumnos como parte del proceso de aprendizaje es la otra práctica docente que la comunidad DBER encontró que es determinante para la mejora del aprendizaje.

En la práctica educativa descrita en esta memoria se fomentó la colaboración entre los alumnos en todas las etapas tanto para diseñar soluciones como para implementarlas en el código de los programas que se descargaban en los robots *GiggleBot*. Esta colaboración entre alumnos se producía en diversos ámbitos: dentro del laboratorio en las horas dedicadas a la realización de la práctica, remotamente a través del foro social del Aula Virtual, y a través de canales externos como el grupo de Whatsapp que mantienen los alumnos.

Sin embargo, al diseñar esta práctica educativa se consideró que había que incidir más en el aprendizaje social, para lo cual se diseñó una etapa de la práctica en la que los alumnos tuvieran que colaborar para diseñar estrategias de solución del problema planteado. Tuvieron así que colaborar no sólo en la búsqueda de soluciones, sino en el proceso de alcanzar un consenso para que los programas desarrollados para los respectivos robots *GiggleBots* de cada equipo pudiesen colaborar entre sí mediante el intercambio de mensajes vía radio para llevar a cabo la estrategia consensuada en la fase de evaluación de la solución de cada equipo.

La introducción del aprendizaje lúdico grupal en esta etapa de la práctica educativa contribuyó así mismo a que los alumnos estuviesen más motivados para llevar a cabo un aprendizaje social.

Reflexiones sobre los aspectos metodológicos

La aplicación de los resultados de la investigación sobre metodologías docentes que la comunidad DBER ha llevado a cabo en los últimos años ha sido trasladada a la práctica educativa descrita en esta memoria, habiéndose logrado una mejora medible cualitativa y cuantitativamente tal como se describe en esta memoria.

Tenemos la esperanza de que dichos resultados puedan contribuir a que otros colegas puedan mejorar su docencia aprovechando las evidencias proporcionadas en esta memoria.

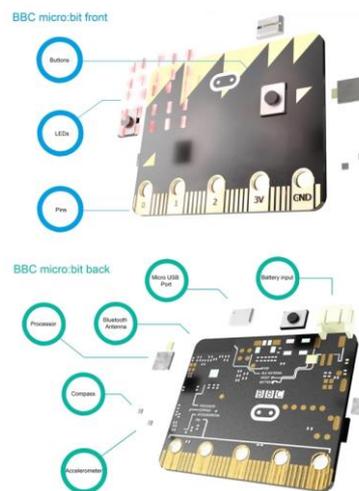
A modo de conclusión sobre los aspectos metodológicos cabe decir que en esta experiencia se corrobora que los resultados de aprendizaje de los alumnos mejoran cuando se utilizan las prácticas docentes consistentes en un énfasis en las prácticas con realimentación precisa a tiempo y en los aprendizajes lúdico y social.

Estos resultados están alineados con otras muchas evidencias evaluadas cuantitativamente en la literatura científica según las cuáles, cuando los estudiantes se involucran en la resolución de tareas no simples que les plantean desafíos intelectuales similares a los que tienen que resolver los expertos de su profesión, en los que hay que aplicar conocimiento experto, y siempre que los estudiantes reciban realimentación detallada a tiempo, los estudiantes aprenden más y obtienen mejores resultados académicos.

Tecnologías o herramientas digitales aplicadas

Habitualmente se supone que es normal que en asignaturas de grados en ingeniería como esta se utilicen tecnologías y herramientas digitales como apoyo a la docencia. Sin embargo, en esta asignatura se ha realizado un esfuerzo especial en este sentido: se ha utilizado un libro multimedia interactivo con simuladores de código y pruebas de evaluación formativa integradas por un lado, y por otro se ha utilizado hardware específico que se ha entregado a los estudiantes para que puedan realizar el aprendizaje práctico de la asignatura. El uso de estas herramientas y tecnologías digitales se ha coordinado en su totalidad desde el entorno del Aula Virtual de la URJC.

1. Plataforma hardware *micro:bit*



El *micro:bit* es una plataforma de hardware diseñada para programar dispositivos para la Internet de las cosas (IoT) que pueden ser utilizados autónomamente para recibir señales de comunicación vía radio o señales captadas por los sensores propios de la placa o por los sensores externos conectados al *micro:bit*, y para emitir señales hacia actuadores como motores de un robot, transmisión de señales de radio, etc. El *micro:bit* es una placa de hardware abierto (*Open Source*), basada en un microcontrolador ARM Cortex-M0 que fue diseñada por la BBC en el Reino Unido para utilizarla en entornos educativos como una plataforma IoT en los colegios ingleses en 2016. Hoy día se utiliza en varios países por todo el mundo, en colegios, institutos y universidades. También en la industria: el Bluetooth SIG utiliza los *micro:bit* para formar desarrolladores en la tecnología [Bluetooth Mesh](#) y la compañía AdaCore los utiliza para formar a desarrolladores en la programación sobre máquina desnuda con el [lenguaje Ada](#).

Incorpora 16KB de RAM y una flash de 256KB, 25 LEDs de pantalla y sensor de luminosidad, conectores de E/S para interacción con otros dispositivos, un acelerómetro, un magnetómetro, un sensor de temperatura, radio Bluetooth y conector USB.

En cuanto al software, tiene instalado un *runtime* programado en C++ por la [Universidad de Lancaster](#). El *runtime* está compuesto por el SW necesario para controlar el HW del *micro:bit*, proporcionando una API de alto nivel en C++ para poder programar más cómodamente el dispositivo. Encima de la API C++ se han desarrollado diferentes entornos de programación en múltiples lenguajes: Python, TypeScript, y varios entornos gráficos de programación con bloques como Scratch 3.0 o MakeCode, este último desarrollado por Microsoft para esta [plataforma](#). Ver <https://microbit.org/code/>. Como alternativa, se puede instalar también el sistema operativo [Zephyr](#) o programar directamente la máquina desnuda, sin ningún runtime, utilizando los lenguajes C/C++, Ada, etc.

En la asignatura hemos utilizado el runtime de la Universidad de Lancaster sobre el que ejecutamos MicroPython, un intérprete de Python para microcontroladores como el ARM Cortex-M0 del *micro:bit*.

2. GiggleBot



GiggleBot es una plataforma robótica diseñada y fabricada por Dexter Industries, compañía americana que lleva 10 años trabajando en robots para entornos educativos, primero para Lego Mindstorms y posteriormente para Raspberry Pi. *GiggleBot* es su primer robot diseñado para *micro:bit*.

El *GiggleBot* es una plataforma robótica que incorpora un zócalo para insertar un *micro:bit* que controle la plataforma. La plataforma *GiggleBot* cuenta con 2 ruedas motorizadas, 9 LEDs RGB, sensores de luminosidad, sensores para seguir líneas, conectores para servos y sensores adicionales. En la asignatura hemos utilizado un sensor adicional de distancias adicional conectado al *GiggleBot* que mediante un láser permite detectar a qué distancia se encuentran los objetos del entorno.

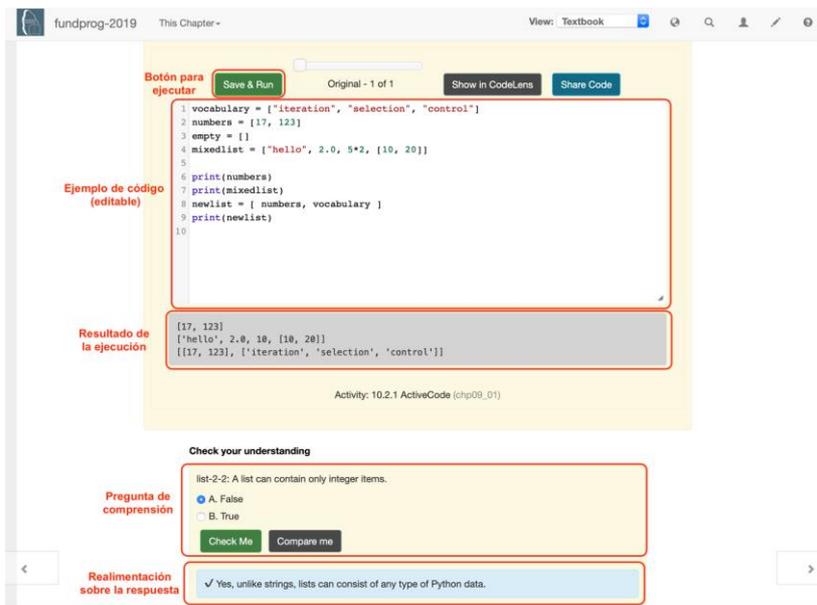
El *GiggleBot* es controlado por el programa que se ejecute en el *micro:bit* insertado en su zócalo. El fabricante proporciona dos módulos de Python como API para acceder a todas sus funciones.

3. Libro multimedia interactivo de Runestone.Academy

[Runestone Academy](#) es un proyecto de Runestone Interactive, fundado en 2011 por Brad Miller, catedrático emérito del Luther College en EE.UU. con aportaciones de muchos otros académicos estadounidenses.

La motivación original del proyecto fue que los libros de texto para *Computer Science* carecen de sentido en la actualidad, fundamentalmente porque los libros de texto son incapaces de ejecutar código. Inspirado en [Skulpt Project](#), un intérprete interactivo de Python que se ejecuta en el navegador, Brad Miller pensó en incorporar un intérprete a su libro de texto sobre Python, que permitiera ejecutar ejemplos y plantear ejercicios de forma interactiva a los estudiantes. Actualmente el portal incluye 18 libros interactivos en el ámbito de *Computer Science*, bajo licencia libre. En nuestra asignatura hemos utilizado el libro *How to Think Like a Computer Scientist*.

El libro interactivo sigue una estructura tradicional basado en capítulos y secciones, pero en cada apartado presenta, entremezclado en el texto explicativo, ejemplos de código que el estudiante puede ejecutar interactivamente para comprobar su salida y preguntas para comprobar su aprendizaje mediante evaluación formativa, así como vídeos explicativos:



The screenshot displays an interactive Python environment. At the top, there is a 'Save & Run' button and a 'Show In CodeLens' button. The code editor contains the following Python code:

```
1 vocabulary = ["iteration", "selection", "control"]
2 numbers = [17, 123]
3 empty = []
4 mixedlist = ["hello", 2.0, 5*2, [10, 20]]
5
6 print(numbers)
7 print(mixedlist)
8 newlist = [ numbers, vocabulary ]
9 print(newlist)
10
```

The output of the code execution is shown below the code editor:

```
[17, 123]
['hello', 2.0, 10, [10, 20]]
[[17, 123], ['iteration', 'selection', 'control']]
```

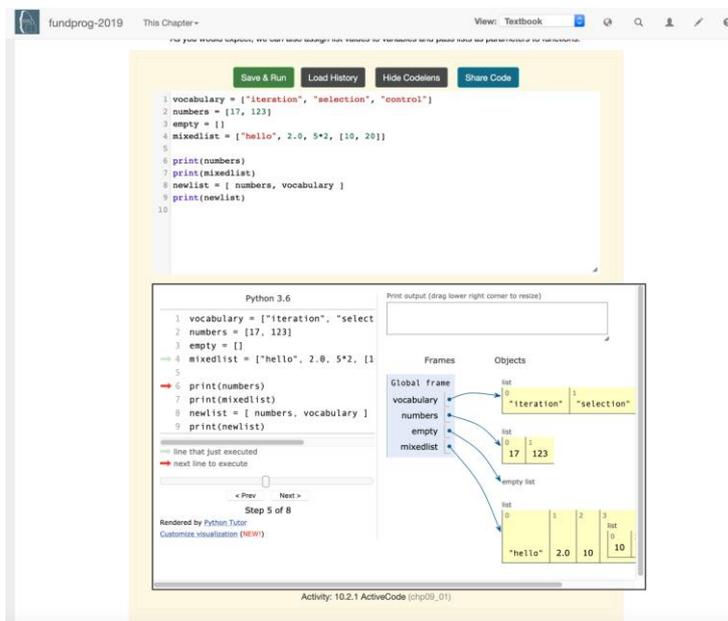
Below the output, there is a 'Check your understanding' section with a question:

list-2-2: A list can contain only integer items.

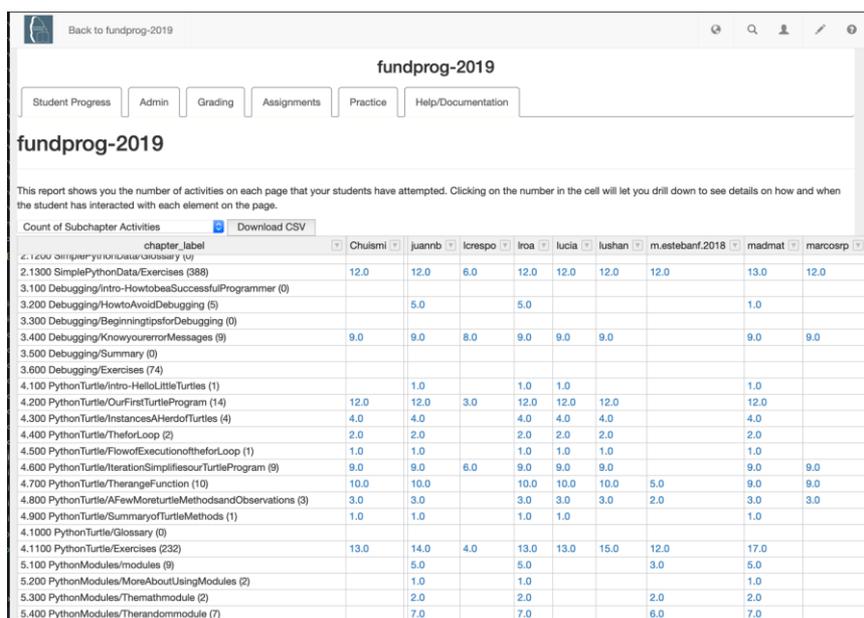
A. False
 B. True

Buttons for 'Check Me' and 'Compare me' are visible. A feedback message at the bottom states: '✓ Yes, unlike strings, lists can consist of any type of Python data.'

Los ejemplos interactivos de código pueden además ser ejecutados paso a paso para comprobar los valores de las variables. En la siguiente figura se muestra una captura de pantalla de un momento en el que el estudiante está ejecutando código interactivamente:



Los estudiantes de nuestra asignatura crean una cuenta en esta plataforma desde el Aula Virtual de la URJC, y a través de ella los profesores podemos comprobar su grado de seguimiento de los temas, añadiendo sus notas a la evaluación de las tareas del el Aula Virtual:



The screenshot shows a report titled "fundprog-2019" with a navigation menu (Student Progress, Admin, Grading, Assignments, Practice, Help/Documentation). Below the title, it states: "This report shows you the number of activities on each page that your students have attempted. Clicking on the number in the cell will let you drill down to see details on how and when the student has interacted with each element on the page." There is a "Download CSV" button.

chapter_label	Chuismi	juannb	lcespo	Iraa	lucia	lushan	m.estebanf.2018	madmat	marcosp
2.1200 SimplePythonData/Exercises (388)	12.0	12.0	6.0	12.0	12.0	12.0	12.0	13.0	12.0
3.100 Debugging/Intro-HowtoBeaSuccessfulProgrammer (0)									
3.200 Debugging/Intro-HowtoAvoidDebugging (5)		5.0		5.0				1.0	
3.300 Debugging/BeginningtipsforDebugging (0)									
3.400 Debugging/KnowyouerrorMessages (9)	9.0	9.0	8.0	9.0	9.0	9.0		9.0	9.0
3.500 Debugging/Summary (0)									
3.600 Debugging/Exercises (74)									
4.100 PythonTurtle/Intro-HelloLittleTurtles (1)		1.0		1.0	1.0			1.0	
4.200 PythonTurtle/OurFirstTurtleProgram (14)	12.0	12.0	3.0	12.0	12.0	12.0		12.0	
4.300 PythonTurtle/InstancesAHerdoofTurtles (4)	4.0	4.0		4.0	4.0	4.0		4.0	
4.400 PythonTurtle/TheforLoop (2)	2.0	2.0		2.0	2.0	2.0		2.0	
4.500 PythonTurtle/FlowofExecutionoftheforLoop (1)	1.0	1.0		1.0	1.0	1.0		1.0	
4.600 PythonTurtle/IterationSimplifiesourTurtleProgram (9)	9.0	9.0	6.0	9.0	9.0	9.0		9.0	9.0
4.700 PythonTurtle/TheRangeFunction (10)	10.0	10.0		10.0	10.0	10.0	5.0	9.0	9.0
4.800 PythonTurtle/AFewMoreTurtleMethodsandObservations (3)	3.0	3.0		3.0	3.0	3.0	2.0	3.0	3.0
4.900 PythonTurtle/SummaryofTurtleMethods (1)	1.0	1.0		1.0	1.0			1.0	
4.1000 PythonTurtle/Glossary (0)									
4.1100 PythonTurtle/Exercises (232)	13.0	14.0	4.0	13.0	13.0	15.0	12.0	17.0	
5.100 PythonModules/modules (9)		5.0		5.0			3.0	5.0	
5.200 PythonModules/MoreAboutUsingModules (2)		1.0		1.0				1.0	
5.300 PythonModules/Themathmodule (2)		2.0		2.0			2.0	2.0	
5.400 PythonModules/Theandommodule (7)		7.0		7.0			6.0	7.0	

Gracias a esta herramienta digital, los profesores abandonamos el ritmo tradicional de “exposición magistral” de los conceptos de la asignatura. Presentamos materiales propios (en castellano) en la clase como motivación para que el estudiante proceda a probar colaborando con sus compañeros los ejemplos interactivos durante la clase que siempre se desarrolla en el laboratorio con un ordenador para cada estudiante. Los materiales propios siguen la estructura del libro interactivo, evaluándose su comprensión mediante pequeñas pruebas de evaluación

- **Etapa 2:** Familiarización con el *GiggleBot* y desarrollo de varios ejercicios para acostumbrarse a la API de sus diversos dispositivos: monitorización a través de sus LCD, movimiento de los motores, orientación por la intensidad lumínica, seguimiento de líneas, medición de distancias, envío y recepción de mensajes por radio.
- **Etapa 3:** Realización del proyecto principal: localización con el *GiggleBot* de varios *micro:bit*-baliza situados en posiciones al azar en un aula, movimiento hacia los mismos (evitando obstáculos) mediante el análisis de la potencia recibida de sus señales de radio, obtención de los “mensajes secretos” emitidos por las balizas, movimiento hacia la zona iluminada del árbitro, y finalmente envío vía radio de los secretos al árbitro una vez alcanzada la zona iluminada, todo ello en el menor tiempo posible.
- **Etapa 4:** Extensión del proyecto principal para mejorar los tiempos de superación de la prueba mediante la colaboración entre los miembros de equipos de 2 ó 3 estudiantes cuyos robots deben repartirse automáticamente las balizas para obtener los secretos en paralelo, transmitírselos entre sí, y finalmente entregar todos los robots todos los secretos al árbitro.

La evaluación de los distintos componentes de estas etapas supone en total un 80% de la evaluación de la asignatura, si bien cada etapa es evaluada por separado. Estas etapas son evaluadas de forma continua a lo largo del periodo lectivo mediante los sistemas descritos a continuación.

1. Evaluación sumativa de los proyectos de la Etapa 1

El estudiante debe entregar a través del aula virtual el código de 3 proyectos de programación planteados en esta etapa, siendo obligatoria la entrega de al menos 2 de ellos.

La rúbrica de evaluación utilizada para cada uno de los proyectos se muestra en la siguiente **Tabla 1**:

	Aprendizaje bajo	Aprendizaje medio	Aprendizaje bueno	Aprendizaje excelente
Ejecución del proyecto	El programa no puede ejecutarse por errores de sintaxis o de ejecución			El programa se ejecuta sin errores de sintaxis o de ejecución
Adaptación a las especificaciones detalladas del proyecto	El programa incumple la mayoría de las especificaciones	El programa incumple algunas de las especificaciones clave, cumpliendo otras	El programa cumple la mayoría de las especificaciones del proyecto, pero incumple alguna no clave	El programa cumple todas las especificaciones del proyecto
Declaración de variables	Nombres de variables inadecuados y variables sin inicializar	Nombres de variables inadecuados o variables sin	Algún nombre de variable inadecuado	Nombres de variables adecuados, variables inicializadas

		inicializar, pero no ambos		
Valores numéricos de las condiciones de ejecución	Valores cableados mediante literales numéricos en el código	Algún valor cableado		Todos los valores declarados en variables/constantes
Legibilidad del código mediante interlineado separado	Todo el código junto sin separar con líneas en blanco	Algunas partes del código mal separadas		Código legible mediante el uso de líneas en blanco
Comentarios en el código	No hay comentarios	Algunos comentarios o cada línea comentada	Algunos comentarios de más en partes que no los precisan	Comentarios adecuados en la partes precisas
Utilización de funciones	Código repetido sin utilizar funciones	Utilización de algunas funciones pero algunas demasiado extensas	Defectos en el paso de parámetros	Uso correcto de funciones y paso de parámetros

El peso de esta evaluación supone un 18% del total de las 4 etapas de la práctica docente completa, y un 15% del total de la asignatura.

2. Evaluación sumativa de los resultados de aprendizaje de la Etapa 1

Además de la evaluación de los proyectos entregados, y posteriormente a la entrega de éstos, los estudiantes realizaron una prueba síncrona individual en el laboratorio en la que se enfrentaron a la realización en 2 horas de 3 programas cortos sobre aspectos desarrollados tanto en los proyectos de esta etapa como del grado de adquisición global de todos los aspectos en los que se había trabajado en la asignatura hasta el momento.

La rúbrica de evaluación de esta prueba se muestra la siguiente **Tabla 2**:

	Aprendizaje bajo	Aprendizaje medio	Aprendizaje bueno	Aprendizaje excelente
Número de ejercicios que cumplen las especificaciones del enunciado	Ningún ejercicio	1 ejercicio	2 ejercicios	3 ejercicios
Utilización de variables	Nombres de variables inadecuados y variables sin inicializar	Nombres de variables inadecuados o variables sin inicializar, pero no ambos	Algún nombre de variable inadecuado	Nombres de variables adecuados, variables inicializadas

Legibilidad general del código	Muy poco legible	Algunas partes/ejercicios poco legibles		Todo el código legible
Utilización de funciones	Código repetido sin utilizar funciones	Utilización de algunas funciones pero algunas demasiado extensas	Defectos en el paso de parámetros	Uso correcto de funciones y paso de parámetros

El peso de esta evaluación supone un 18% del total de las 4 etapas de la práctica docente completa, y un 15% del total de la asignatura.

3. Evaluación sumativa de los ejercicios de la Etapa 2

El estudiante debe entregar a través del aula virtual los ejercicios de la etapa 2.

Para la evaluación del código de estos programas se utilizó la misma rúbrica mostrada en la Tabla 1.

El peso de esta evaluación supone un 4% del total de las 4 etapas de la práctica docente completa, y un 3% del total de la asignatura.

4. Evaluación sumativa del proyecto de la Etapa 3

El estudiante debe entregar a través del aula virtual el código del proyecto desarrollado en la Etapa 3.

Para la evaluación del código se utilizó la misma rúbrica mostrada en la Tabla 1.

Adicionalmente, se celebra en un laboratorio docente de la universidad, acondicionado específicamente para la ocasión, una prueba de campo conjunta competitiva de los proyectos de los estudiantes, ejecutando su código sobre sus robots.

Para la evaluación de esta prueba de campo en el laboratorio docente se utilizó la rúbrica que aparece en la siguiente Tabla 3:

	Aprendizaje bajo	Aprendizaje medio	Aprendizaje bueno	Aprendizaje excelente
Localización y navegación (3 balizas)	Ninguna	1 baliza	2 balizas	3 balizas
Localización y navegación árbitro	No localiza la zona iluminada del árbitro	Alcanza la zona del árbitro pero no logra detenerse en zona de entrega de resultados		Entrega correcta de resultados al árbitro

Evitación de obstáculos en la navegación	No implementada	Se atasca en algún obstáculo	Supera los obstáculos pero pierde la orientación a la baliza	Supera los obstáculos sin perder la orientación
Velocidad de la ejecución de la prueba	El robot no llega a completar la prueba	El robot completa la prueba en un tiempo significativamente superior al de la mayoría	El robot completa la prueba en un tiempo similar al de la mayoría	El robot completa la prueba en un tiempo significativamente inferior al de la mayoría
Reacción del estudiante a los problemas imprevistos en su programa	El estudiante abandona sin intentar corregir su programa		El estudiante corrige su programa en un plazo largo de tiempo	El estudiante corrige su programa en un plazo breve de tiempo o bien su código no sufre problemas imprevistos

El peso de esta evaluación supone un 21% del total de las 4 etapas de la práctica docente completa, y un 17% del total de la asignatura.

5. Evaluación sumativa del proyecto de la Etapa 4

El estudiante debe entregar a través del aula virtual el código del proyecto desarrollado en la Etapa 4.

Para la evaluación del código se utilizó la misma rúbrica mostrada en la Tabla 1.

Adicionalmente, se celebra en el laboratorio docente de la universidad acondicionado específicamente para la ocasión una prueba de campo conjunta competitiva por equipos de los proyectos de los estudiantes, ejecutando su código sobre sus robots.

Para la evaluación de esta prueba de campo se utilizó la rúbrica que aparece en la siguiente Tabla 4:

	Aprendizaje bajo	Aprendizaje medio	Aprendizaje bueno	Aprendizaje excelente
Explicación oral de la estrategia de reparto del trabajo entre los miembros del equipo	Explicación confusa o no concordante con la que dan otros miembros del equipo			Explicación clara y con soltura de la estrategia empleada

Reparto equitativo de tareas entre los miembros del equipo			1 robot organiza y 1 o 2 realizan el trabajo	Todos los robots realizan un trabajo similar
Velocidad de la ejecución de la prueba	El equipo no llega a completar la prueba	El equipo completa la prueba en un tiempo significativamente superior al de la mayoría	El equipo completa la prueba en un tiempo similar al de la mayoría	El equipo completa la prueba en un tiempo significativamente inferior al de la mayoría
Reacción del equipo de estudiantes a los problemas imprevistos en su programa	El equipo abandona sin intentar corregir su programa		El equipo corrige su programa en un plazo largo de tiempo	El equipo corrige su programa en un plazo breve de tiempo o bien su código no sufre problemas imprevistos

El peso de esta evaluación supone un 12.5% del total de las 4 etapas de la práctica docente completa, y un 10% del total de la asignatura.

6. Evaluación sumativa de los resultados de aprendizaje de las Etapas 2 a 4

Además de la evaluación de los proyectos entregados, y posteriormente a la entrega de éstos, los estudiantes realizaron una prueba síncrona individual en el laboratorio en la que se durante 3 horas tuvieron que realizar 3 programas para evaluar el aprendizaje llevado a cabo en los proyectos de esta etapa, y para evaluar la adquisición global de conocimientos llevada a cabo en la asignatura hasta el momento.

Para la evaluación del código de estos programas se utilizó la misma rúbrica mostrada en la Tabla 2.

El peso de esta evaluación supone un 25% del total de las 4 etapas de la práctica docente completa, y un 20% del total de la asignatura.

7. Evaluación formativa de otras partes de la asignatura

Aunque no son objeto del análisis de esta práctica docente basada en *micro:bit* y *GiggleBot*, queremos señalar que otros aspectos de la evaluación formativa como el trabajo en los ejercicios del libro interactivo, y la participación en clase y en el foro de la asignatura, son objeto de otro 20% de la evaluación de la asignatura.

4. Resultados

Metodología de análisis

Esta práctica docente es una evolución en profundidad de la práctica que ya se realizó en el curso anterior, extendiendo y ampliando las metodologías aplicadas por un lado, y mejorando la plataforma de prácticas con nuevo hardware: 2 *micro:bit* en lugar de 1 por estudiante y 1 robot por cada estudiante.

La asignatura que sucede a esta en el segundo cuatrimestre del estudiante, “Algoritmos y Estructuras de Datos”, depende fuertemente de la madurez que alcancen los estudiantes en los resultados de aprendizaje de ésta que la precede. Intencionadamente, en el curso 19-20 decidimos no realizar cambios metodológicos en la segunda asignatura para poder comprobar así los efectos de los cambios en esta asignatura.

Por ello hemos realizado un análisis cuantitativo de los resultados académicos de los estudiantes tanto en esta asignatura como en la que la sucede. Así mismo hemos realizado una encuesta a los estudiantes con preguntas específicas sobre la experiencia de esta práctica docente.

Resultados

1. Análisis de los resultados académicos en esta asignatura

Los resultados académicos han sido muy similares en cuanto a números de estudiantes calificados como Sobresaliente, Notable o Aprobado. Sin embargo, hemos detectado efectos significativos en los apartados que se recogen en la siguiente Tabla 5:

	Curso 18-19	Curso 19-20
Estudiantes que completan la evaluación continua	85%	90%
Estudiantes que entregan última etapa proyecto final	19%	40%
Estudiantes calificados en el primer cuartil proyecto final	16%	33%

Atribuimos al efecto motivador de las metodologías y tecnologías aplicadas en este curso académico el que se haya duplicado el porcentaje de estudiantes que entregan la última etapa del proyecto final, y el que el porcentaje de estudiantes que reciben una calificación por su proyecto en el primer cuartil se haya incrementado de manera tan significativa.

2. Análisis de los resultados académicos en la asignatura que la sucede

La asignatura “Algoritmos y Estructuras de Datos” está situada en el segundo cuatrimestre del primer curso del grado. Es fundamental para la segunda asignatura la robustez con la que los estudiantes alcanzan los resultados de aprendizaje en la primera asignatura objeto de este análisis. En el precedente curso 18-19 los profesores de la asignatura detectaron una falta de madurez en la formación de los estudiantes a la entrada de esa segunda asignatura: si bien los estudiantes habían alcanzado los resultados de aprendizaje previsto, daba la sensación de que no lo habían hecho con la suficiente robustez para poder afrontar con soltura la segunda.

Por ello se implantaron en la primera asignatura los cambios metodológicos descritos en esta propuesta, con el propósito de aumentar la motivación de los estudiantes con las metodologías y tecnologías empleadas, de forma que dedicaran más tiempo a profundizar en sus resultados de aprendizaje.

Así, hemos analizado comparativamente los resultados en estos dos últimos cursos en esta segunda asignatura, cuyos datos se reflejan en la siguiente Tabla 6:

	Curso 18-19	Curso 19-20
Sobresalientes y Matrículas de Honor	2%	2%
Notables	13%	21%
Aprobados	18%	42%
Suspensos y No presentados	60%	33%

Puede comprobarse como la mejora de los resultados ha sido muy significativa, duplicándose en porcentaje los notables y aprobados, y habiéndose reducido a la mitad los suspensos.

3. Análisis de la encuesta de satisfacción con la asignatura

Para profundizar en el análisis de los resultados obtenidos con los cambios en la asignatura, hemos pasado una encuesta a los estudiantes para la que hemos utilizado preguntas del *Student Evaluation of Educational Quality*

Questionnaire (SEEQ) [7]. Además, hemos decidido pasar la encuesta diferida en el tiempo respecto a la finalización de la asignatura, de acuerdo con los estudios que señalan que los beneficios de haber completado una asignatura pueden no ser inmediatamente evidentes para los estudiantes, sino que se revelan un tiempo después, cuando esos beneficios han podido ser observados por el alumno en otras asignaturas [8].

Puede consultarse la encuesta completa en el sitio web referido en el Resumen. La siguiente Tabla 7 extrae los resultados más relevantes en relación con la práctica docente objeto de esta propuesta:

Estudiantes que valoran “poco o nada interesantes” prácticas SIN <i>micro:bit+GiggleBot</i>	58%
Estudiantes que valoran “bastante o muy interesantes” prácticas SIN <i>micro:bit+GiggleBot</i>	11%
Estudiantes que valoran “poco o nada interesantes” prácticas CON <i>micro:bit+GiggleBot</i>	11%
Estudiantes que valoran “bastante o muy interesantes” prácticas CON <i>micro:bit+GiggleBot</i>	84%
Estudiantes muy en desacuerdo o en desacuerdo con que las prácticas con <i>micro:bit+GiggleBot</i> “son una parte muy valiosa de la asignatura”	11%
Estudiantes de acuerdo o muy de acuerdo con que las prácticas con <i>micro:bit+GiggleBot</i> “son una parte muy valiosa de la asignatura”	79%

Puede comprobarse cómo los estudiantes aprecian de manera muy destacada esta práctica frente a otras más convencionales, y lo consideran muy mayoritariamente una parte “muy valiosa” de la asignatura.

Por último, incluimos aquí los comentarios de texto libre de los estudiantes sobre esta práctica docente:

- “Tienen un componente material y visual interesante, además puedes ver funcionalidad directa fuera de una pantalla o código siendo capaz de observar cómo se interactúa con el mundo físico (movimiento del *GiggleBot*, transmisión de señales con el microbit, etc.)”
- “Porque es interesante empezar a aplicarlo en hardware”
- “Me gustó trabajar con algo físico y ver su funcionamiento”
- “Era una forma de ver los efectos que tenían las cosas que programabas en el microbit en la vida real y para alguien que nunca ha programado esta bien.”

- “Ves tu trabajo reflejado en algo ‘físico’”
- “Me parece que es algo que venía buscando al entrar en esta carrera así que está bien usar eso para aprender programación”
- “Cambiaba la plataforma sobre la que estábamos acostumbrados a trabajar (turtle), se sentía como una novedad. Además, el proyecto de las balizas fue muy interactivo. Me gustó el tener que trabajar en equipo con otros 2 compañeros.”
- “Me gustó la idea de manejar un robot desde el principio la verdad”
- “Porque era una forma de aplicar la programación a algo físico, y ver que podíamos programar un robot me hacía sentir bien”
- “Creo que estaba bien hilado con el grado que estamos estudiando y fue una práctica en la que se mezcló programación pero también algo tangible.”
- “Me parecieron bastante interesantes y daban un toque más práctico a la asignatura”

4. Evaluación de la calidad de la enseñanza basada en estándares

La aplicación de nuevas metodologías debe evaluarse sistemáticamente observando los resultados académicos, por un lado, y analizando las prácticas docentes. Aplicar una metodología más o menos de moda no es suficiente para evaluar la actividad docente.

Por ello, el cambio metodológico introducido en esta asignatura ha sido acompañado no sólo de la evaluación de los resultados académicos y de encuestas a los estudiantes, sino también por una evaluación de la calidad de la enseñanza basada en estándares. Se proporciona a modo de resumen de esta evaluación basada en estándares el resultado de la encuesta *Teaching Practices Inventory (TPI)* aplicada a esta asignatura. Esta encuesta está diseñada por los profesores Sarah Gilbert y Carl Wieman, miembros destacados de la comunidad DBER. Los resultados pueden verse en el sitio web referenciado en el Resumen de esta memoria.

Esta encuesta permite comparar diferentes prácticas docentes al recoger muchas de las decisiones que un profesor tiene que adoptar al diseñar y enseñar una asignatura, y proporciona una caracterización objetiva y detallada de la mayoría de las prácticas docentes.

Conclusiones

El efecto motivador de esta práctica docente ha mejorado significativamente los resultados académicos de esta asignatura en el curso académico en que se ha puesto en marcha. Además, esta notable mejora en los resultados de aprendizaje adquiridos por los alumnos se ha puesto de manifiesto en el nivel con que los alumnos afrontan la asignatura que sucede a esta en su plan de estudios. En esta última la mejora de resultados académicos ha sido especialmente notable, siendo destacable una reducción muy notable en el número de alumnos que abandonan la segunda asignatura en las primeras semanas del cuatrimestre.

Las encuestas realizadas a los estudiantes muestran cómo éstos aprecian, de manera muy destacada, esta práctica docente por encima de otras más convencionales, y lo consideran muy mayoritariamente una parte “muy valiosa” de la asignatura, realizando comentarios muy entusiastas sobre la misma.

Referencias

- [1] Davi Bernardo Silva, Rafael de Lima Aguiar, Diogo Steinke Dvconlo, Carlos N. Silla. Recent Studies About Teaching Algorithms (CS1) and Data Structures (CS2) for Computer Science Students. IEEE Frontiers in Education Conference (FIE), 2019
 - [2] Lauro André Ribeiro, Thaísa Leal da Silva, Andréa Quadrado Mussi. Gamification: a Methodology to Motivate Engagement and Participation in a Higher Education Environment. International Journal of Education and Research, 6(4), 2018
 - [3] JR Savery. Overview of problem-based learning: Definitions and distinctions. Essential readings in problem-based learning, 2015
 - [4] Crouch, Catherine H. and Mazur, Eric. Peer Instruction: Ten Years of Experience and Results. American Journal of Physics, 69(9), 2001
 - [5] Wieman, C. E. Why not try a scientific approach to science education? Change, 39(5), 2007
 - [6] Brad Miller, David Ranum, Jeffrey Elkner, Peter Wentworth, Allen B. Downey, Chris Meyers, and Dario Mitchell. How to Think Like a Computer Scientist. Runestone Academy Interactive edition, 2016.
 - [7] Marsh, HW. SEEQ: A reliable, valid, and useful instrument for collecting students' evaluations of university teaching. British Journal of Educational Psychology, 52(1), 1982, 77–95.
- Richardson, John. Instruments for obtaining student feedback: A review of the literature. Assessment & Evaluation in Higher Education Vol. 30, No. 4, August 2005, 387–415.

5. Equipo docente



José Centeno González

Ingeniero de Telecomunicación por la Universidad Politécnica de Madrid, y Doctor por la misma universidad. Profesor Titular de la URJC en el DTSCSTC en la ETSIT. Pertenece al grupo GSyC desde 1993. Junto con Pedro de las Heras y otros profesores pusieron en marcha en la UC3M laboratorios docentes basados en NetBSD y GNU/Linux, incluyendo su diseño, configuración y administración, y en 2002 trasladaron el modelo a la URJC. Desde 2002 utiliza LMS como soporte a todas sus asignaturas, con diversos sistemas hasta desembocar en Moodle, participando en el piloto de Moodle de la URJC. Desde 2005 ha utilizado en sus asignaturas herramientas de aprendizaje activo y PBL. Ha participado en la elaboración y revisión de planes de estudios de carreras en el ámbito de las TIC desde 2003, y continua en la actualidad. Todos los materiales académicos de todas sus asignaturas siempre han estado disponibles bajo licencias libres, Creative Commons desde su creación. Ha obtenido evaluaciones favorables en todas las convocatorias de Docencia, incluyendo 2 notables y 1 excelente. Ha participado en proyectos de innovación docente desde 2006, recibiendo en 2017 un Premio en Propuestas de Innovación Docente en los IV Premios Profesores Innovadores de la URJC.



Pedro de las Heras Quirós

Licenciado en informática y doctor en informática por la Universidad Politécnica de Madrid, desde 1994 ha sido profesor en asignaturas de grado y postgrado en diversas titulaciones de ingeniería en la Universidad Carlos III de Madrid y en la Universidad Rey Juan Carlos (URJC). Actualmente es profesor titular de universidad del departamento de Teoría de la Señal y las Comunicaciones y Sistemas Telemáticos y Computación en la URJC. Junto a la profesora Eva M. Castro Barbero recibió en 2018 el primer premio en los V Premios Profesores Innovadores por la práctica educativa 'Uso de laboratorios portables de redes de ordenadores para el aprendizaje de protocolos de comunicaciones SDN (Software Defined Networking), mediante tecnología PBL. Junto al profesor José Centeno González y otros ha desarrollado una labor continuada de promoción y soporte de la docencia práctica en ingeniería.