Received: 28 November 2016 | Accepted: 17 March 2017
DOI: 10.1002/cae.21827

RESEARCH ARTICLE

WILEY

Improving student engagement on programming using app development with Android devices

César Fernández[1] | María Asunción Vicente[1] | M. Mar Galotto[2] | Miguel Martinez-Rach[2] | Alejandro Pomares[2]

[1] Department of Systems Engineering and Automation, Miguel Hernández University, Elche, Spain

[2] Department of Physics and Computer Science, Miguel Hernández University, Elche, Spain

Correspondence
Dr. María Asuncion Vicente, Avenida de la Universidad s/n, Edificio Innova, Elche, Alicante, Spain 03202.
Email: suni@umh.es

Abstract

This work presents our experience on Android teaching at Miguel Hernández University (Elche, Spain). We decided to orientate our courses toward Android app development, and encouraged students to carry their Android phones or tablets to the classroom. The results, in terms of student motivation, satisfaction and engagement in programming have been extraordinary.

KEYWORDS

Android, app development, coding, engineering education, student motivation

1 | INTRODUCTION

Increasing student motivation to programming subjects in Engineering and Computer Science degrees has been a key issue in Miguel Hernández University (MHU) since its creation in 1997. We have been equipping our classrooms with different devices, mainly robotic devices, with successful results. Nowadays, however, such hardware is no longer necessary. The mobile devices are now everywhere and the task of developing apps is highly motivating for most of the Engineering students. In this paper, we present our experience in Android and Java teaching at MHU (Elche, Spain). Since 2012, we decided to orientate some of our courses toward Android app development, and encouraged students to carry their Android smartphones or tablets to the classroom. The results, in terms of student motivation, satisfaction and engagement, have been really positive.

Object-oriented programming languages and, in general, computer programming or coding are dry, difficult subjects for some students [15]. Many of them do not get the necessary motivation to keep on studying these subjects. However, mastering a programming language is a necessary skill for future engineers. Moreover, coding is inherent in many subjects of the Engineering and Computer Science degrees [5], from the most basic courses, where the students usually learn languages as C, C++ or Java on console applications, to advanced courses as Robotics, Microcontrollers or Control Theory, where different languages can be used as programming tools (Matlab, assembler, Python, etc.) in combination with lab devices. Our experience in teaching such subjects [6,22,23] shows that students are discouraged when delving deeper in the programming language features and finally, they are only able to learn their basics.

In order to increase student motivation, different alternatives have been used for the last years. They can be roughly classified in four groups:

- To equip the classroom electronic devices and robots [3]. The students find programming a much more motivating task when their programs can make a robot move, walk, or even become intelligent. Obviously, the main drawback of this alternative is the cost of the equipment, which cannot always be afforded.
- The use of remote laboratories [8]. Remote laboratories help in keeping costs low, as multiple students, which operate it through the Internet, can share a single device. There are several disadvantages, though: concurrency of several students at the same time, devices that need human

supervision (ie, walking robots that may fall, devices that need charging, etc.).

- The use of virtual laboratories [2]. Cost is kept to a minimum, as there are no devices at all. Concurrency is not a problem either, as multiple students can be connected and performing experiments at the same time. However, it becomes difficult to motivate students when they work with simulations instead of real devices.
- Another alternative is to design the practice sessions so that they result interesting to the students, using just the computers and not needing additional hardware; see, for example, [19,21]. However, such alternative will not be covered in the present paper, where the focus will be set on virtual or real devices.

Nowadays, however, it is not necessary to equip the classrooms with such hardware. The students own a much more powerful and versatile hardware: their mobile devices. Students get highly motivated when their programs become an application for their own telephones or tablets.

The structure of the paper is as follows: first, we will present our past educational experiences with classroom devices, remote laboratories and virtual laboratories. Then, we will describe our recent experience in Android and Java programming focused on Android app development; and we will show the results obtained. Finally, we will draw some conclusions.

2 | PAST EDUCATIONAL EXPERIENCE WITH CLASSROOM DEVICES: ROBOTS AND SPECIFIC EQUIPMENT

For years, in our Engineering degrees at MHU, we have tried to increase student motivation by equipping our classrooms with different devices, mainly robotic devices. The results have been highly positive.

Our first experience introducing robots in the classroom started in 1999 with the use of *Rug Warrior* robots [13] in the *Real Time Computer Systems* subject, belonging to the Industrial Engineering degree of MHU. Apart from the regular practice sessions carried out during the course, the best students presented their projects in a final exhibition. The devices were versatile and robust. Later on, in 2001, we added *SoccerBot* robots (http://www.joker-robotics.com/eyebot/socbot.html) for the same subject.

In 2007 we began to use *Moway* robots (http://moway-robot.com/) as autonomous mobile robots to teach microcontroller programming and collaborative robotics. Compared to *Rug Warrior* or *SoccerBot*, *Moway* is much more compact and easier to connect to the computer. Also, one of its great advantages is its low cost, allowing several units in the same classroom. The earlier Moway versions could be programmed just in C or PIC assembler, but now it is possible to program using a much intuitive software based on flowcharts.

A completely different approach is the use of *Robonova* [10] or similar walking and dancing robots. They are more costly, so it is not feasible to equip a complete classroom. Instead, only one or two units are usually bought. We used them mainly for final degree projects. Figure 1 shows some the mobile robots used by the students.

Apart from robots, we have introduced in our classrooms microcontroller development boards for assembler programming learning. In 2000, we started using the educational development board *Handy Board* [17] based on the Motorola 68HC11, one of the most popular microcontrollers based on Von Neumann architecture.

Soon, the Harvard architecture swept the microcontroller market, and specifically the family of Microchip PIC microcontrollers was expanded to become one of the market leaders. In this situation, during 2003, we decided to introduce in the classroom new programming practices based on the educational development board *MicroPIC Trainer* [18]. The *MicroPIC Trainer* is based on Mid-Range family PIC microcontrollers: 16F84X and 16F87X, it allows PIC microcontrollers to be interfaced with external circuits and peripheral devices.

For two academic years, both development boards (the Handy Board based on Motorola 68HC11 and the MicroPIC Trainer based on PIC16F8XX) were used simultaneously by the Engineering students, allowing students to learn the particularities of both microcontroller architectures.

In 2006, we adopted a new PIC development board: *EasyPIC* (http://www.mikroe.com/easypic/) version 4, which includes more peripheral devices than the previous board. The current version of EasyPIC is 7 [4], and it is equipped with more modern peripheral devices and is able to use several models of PIC microcontrollers. Figure 2 shows the external appearance of some of the microcontroller development boards described above.

Throughout these years, at the Engineering degrees of MHU, we have seen how students engage more in microcontroller programming practices if such development boards are used instead of using just the development programming environment with emulators (MPLAB in the case of PIC microcontrollers). However, it is almost unaffordable to have in the classroom one development board for each student, so they must work in groups and share the equipment. Similarly, robot programming by students encourages their interest in learning advanced programming, however, maintaining a classroom with robots has some serious problems: first, the robots have a fairly intrinsically high cost and secondly, the rapid technological changes makes them obsolete within a few years of use.

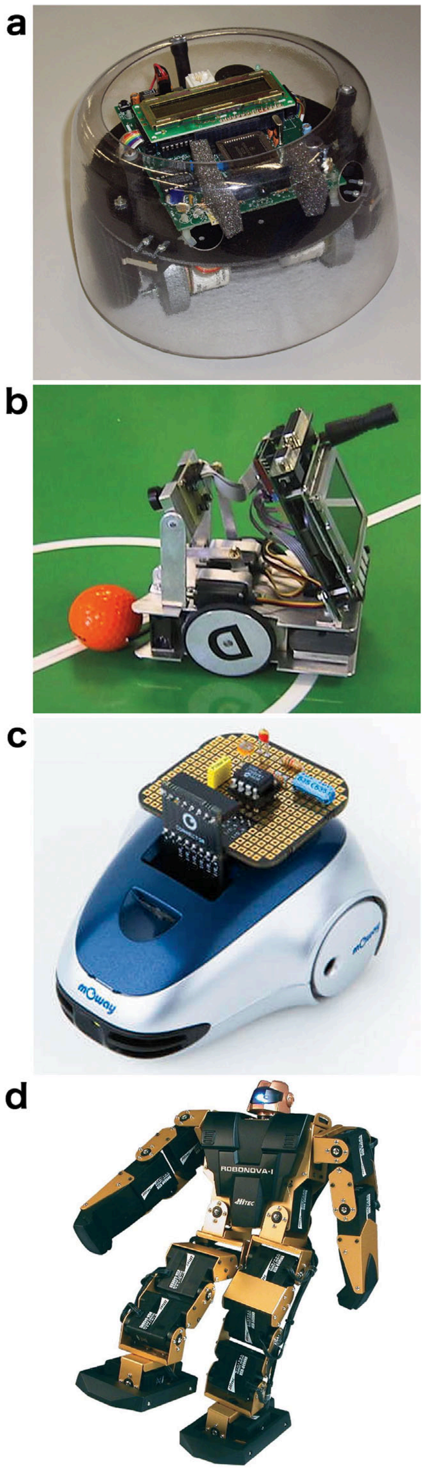


FIGURE 1 Some of the mobile robots used by the students: (a) Rug Warrior. (b) SoccerBot. (c) Moway. (d) Robonova

3 | PAST EDUCATIONAL EXPERIENCE WITH REMOTE AND VIRTUAL LABORATORIES

Concerning remote laboratories, we have used them for training in different control theory subjects, both in Industrial Engineering and Telecommunications Engineering degrees. The main equipment used by students in these subjects was a DC servomotor, namely the Feedback 33–100 model. Although the classroom was equipped with enough units for all the students, we developed a remote lab so that students could also perform experiments with the real device from their homes. More details about this remote control laboratory, named Recolab, can be found in [12].

In the field of control theory, we have also created purely virtual laboratories. In [7] we present two virtual experiments: magnetic levitation control for a train and speed control for a car. They were developed as Matlab and Simulink applications, respectively.

The use of remote and virtual laboratories helps in keeping costs low, but they have numerous disadvantages, especially on remote laboratories where the concurrency of

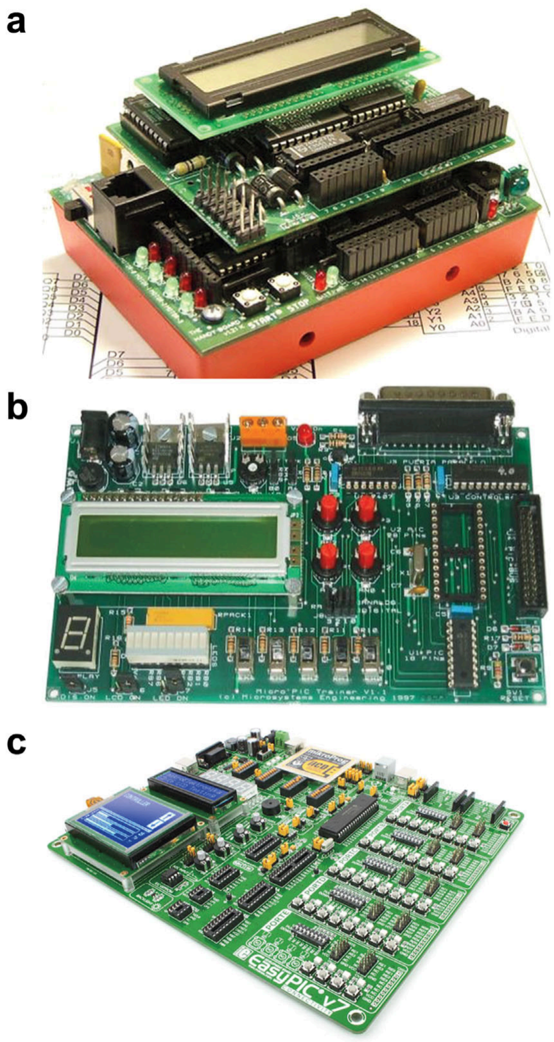


FIGURE 2 We have used in our classrooms microcontroller development boards, these are the more representative: (a) Handyboard. (b) MicroPIC Trainer. (c) EasyPIC v7

several students at the same time might produce severe delays, devices that need human supervision might block the process. With the use of virtual laboratories, the costs are even lower and there are not problems with the equipment, except for the performance of the student computer, however, students do not feel attracted when they work with simulations instead of real devices. Often the students tend to confuse concepts and mix the virtualization processes with the physical system that is being analyzed.

4 | STUDENT'S EVALUATION OF THE PROGRAMMING COURSES AT MHU

Feedback from the students' experience on the programming courses at MHU is obtained through an anonymous survey. This questionnaire consists of different questions related to the organization and management of the course, the course contents and material, as well as the instructor's teaching ability.

Table 1 shows some results of the statistics made with the results of these surveys on the courses where programming is relevant. The courses have been classified into three types according to the equipment used in the lessons: (1) only computer; (2) electronic equipment or robots; (3) remote or virtual laboratories. In addition, for each course the programming language studied is indicated. The numerical data show the average value of the students' assessment of each question in the survey. The table only shows the most relevant questions, related to contents, methodology, material, format, usefulness, and interactivity of the lessons. The data have been obtained in the surveys carried out from 1997 to 2012. Some subjects were evaluated every academic year and others every 2 years. Therefore, all mean values correspond to seven or more editions of the same programming course. In section 6 of this paper, we analyze in detail the data presented in this table.

5 | RECENT EDUCATIONAL EXPERIENCE WITH MOBILE DEVICES BASED ON ANDROID

Over the last few years, the emergence of smartphones and the spread of Android devices have led us to a change in our strategy. For certain subjects, like Java programming, it is no longer necessary to equip the classrooms with robots or other devices. And virtual or remote laboratories are not needed either. Students get extraordinarily motivated when their programming lessons become applications for their own telephones or tablets [9,14].

In brief, our strategy on Android programming learning is simple: first, we orientate our Java programming courses toward Android application development; second, we encourage students to bring their Android phones or tablets to the classroom; third, every lesson given is used to create or improve an Android app that students develop, install and test in their own devices. Obviously, some students do not own an Android device. They can follow the classes running the applications in a simulator of their choice.

More in detail, we decided to create two courses: a *basic course* to start from scratch on app development and an *advanced course* to learn Java programming.

- The first course, "Android programming" was open to all students with no prerequisites. During the course, the students developed simple Android applications (quiz games, sensor reading apps, etc.) with as few lines of Java code as possible. The main focus was set on interface design, application structure and usage of Android resources. The goal of this first course was to motivate students in learning Java to improve their apps.
- The second course, "Java programming for Android," was offered as a continuation of "Android programming." The contents covered all the basics of Java language, with Android examples that students tested on their own devices.

Both courses are described in more detail in the next sections.

5.1 | Android programming course

This introductory course was structured in four sessions of 5 hr each. The concepts addressed were:

- Android platform specifications.
- Comparison with other platforms (iOS/Windows Phone/ Symbian).
- Android architecture.
- User interface design.
- Sensor and input management.
- Apps with multiple activities.
- Adding ads to an application.
- Signing the application.
- Publishing apps in Google Play.

In this introductory course to Android programming, theoretical explanations of the previous Android concepts are combined with practical programming projects where simple apps developed by the students are tested directly on their personal smartphones and tablets. In the Android projects of these first apps, the use of Java code is kept to a minimum, but the learning objective is delving into aspects of the design of the user interface (XML) and managing the smartphone sensors. Figure 3 shows the main screen of two of the apps developed in this course: (a) shows a very simple app built

TABLE 1 Course evaluation statistics of the engineering courses at MH University where coding is a fundamental competence

Name	Type	Language	#4 methodology	#5 contents	#6 material	#7 format	#10 usefulness	#11 interactivity
Computer basics	1	C	3.57	3.78	4.12	4.25	3.30	3.92
Applied informatics	1	C++	3.22	3.67	3.54	3.98	3.08	3.52
Systems theory	1	Matlab	4.08	4.17	4.20	4.25	3.76	4.07
Artificial intelligence and pattern recognition	1	Matlab	3.98	4.74	4.33	4.65	4.53	3.14
Real time computer systems	2	C	4.15	4.33	3.87	4.10	4.08	3.13
PLCs and control systems	2	STEP, Matlab	4.33	3.56	3.83	3.78	4.12	4.56
Electronic and automatic systems	2	Assembler, C	4.35	4.89	4.21	4.28	4.77	4.11
Advanced control	2	Matlab	3.67	3.87	4.10	3.86	3.22	3.75
Industrial computer systems	2	STEP, Java	4.67	4.56	4.47	4.46	3.84	4.78
Industrial automation	2	STEP, Matlab	4.35	4.02	4.08	4.40	3.90	4.22
Manufacturing technology and machines	3	G&M (CNC)	2.98	3.21	2.78	3.75	2.87	3.02
Control of robots and sensory systems	3	Matlab	4.19	3.87	4.03	3.87	4.13	4.18
Robotics and computer vision	3	Matlab	4.01	3.81	3.78	4.05	4.45	4.16
Computer assisted manufacturing	3	G&M (CNC)	2.67	2.86	3.10	2.56	2.10	2.99

Evaluation scale: 1 = very weak, 5 = excellent.
Item #4: The methodology used in the course. Item #5: The adequacy of the contents. Item #6: The material delivered: presentation, source code, videos, tutorials and references. Item #7: The format of the lessons: face to face, online, mixed. Item #10: Trainers provide practical and useful examples for their career. Item #11: Trainers interact with the group (level of participation, listening, asking questions. . .).
Data obtained during the courses' editions from 1997 to 2012.

from the SDK objects: *AnalogClock* and *DigitalClock*; while (b) shows the main screen of a more sophisticated app where several buttons produce different actions as open a navigator, change a text, read a text using the *TextToSpeech* resource, etc.

The first session of the course, called "Installing the Android development environment," is focused on the installation procedure for the development environment. Initially, in 2012, Eclipse was used together with the Android SDK and recently, we are using the new platform Android Studio. The second session of the course is aimed at designing and creating user interfaces, focusing on the various properties of the objects managed from layout files in XML. In the third session, the objective is managing inputs and smartphone sensors, so that several simple apps able to manage the camera, accelerometer, proximity sensor and GPS are made by the students. On the last day of this intensive course, students learn to combine multiple activities in a same app and the procedure to follow to publish their apps on the Google Play store distribution.

Most students attending this first course are very pleased with the results achieved in such a short time. They discover that just in a few hours of dedication they have been able to develop many apps, and that helps a lot in increasing their interest to deepen in the study of Java language.

5.2 | Java programming for Android course

The second course was structured in six sessions of 4 hr each. Apart from these sessions, students had to fulfill 6 extra hours of supervised personal work. The concepts addressed were:

- Java language specifications.
- Object oriented programming.
- Java basics: expressions, variables, types, and operators.
- Java objects.
- Arrays, lists, and loops.
- Creating classes and methods.
- Interfaces.
- Java class library.
- Android SDK.
- OS objects: android.os, android.provider, android.app, android.content
- UI objects: android.view, android.text, android.widget

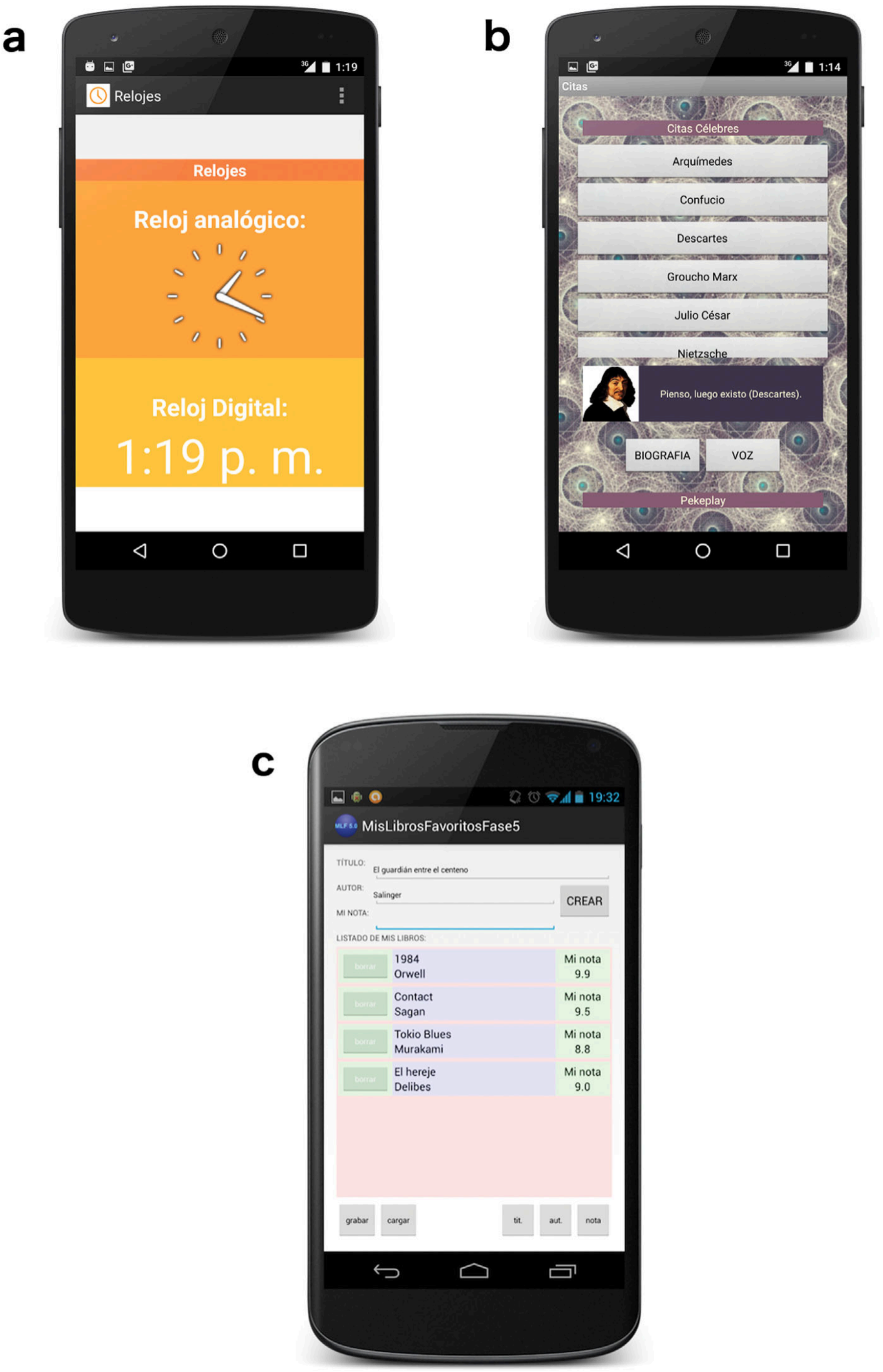


FIGURE 3 (a) Screenshot of one of the basic apps, from the Android programming course. (b) Screenshot of a more complex app, from the Android programming course. (c) Screenshot from an app focused on database management, from the course of Java programming for Android

- Other objects: android.util, android.media
- Practical examples using Android SDK on Eclipse platform.

The concept list is similar to a standard Java course, but students were extraordinarily motivated right from the first lesson. They knew they were learning a tool to improve the apps they had developed in the previous, introductory course.

Most of the Java topics covered in the course were used for an Android application that students created in the first lessons and upgraded during the course. The app was a simple book database that grows following this schema:

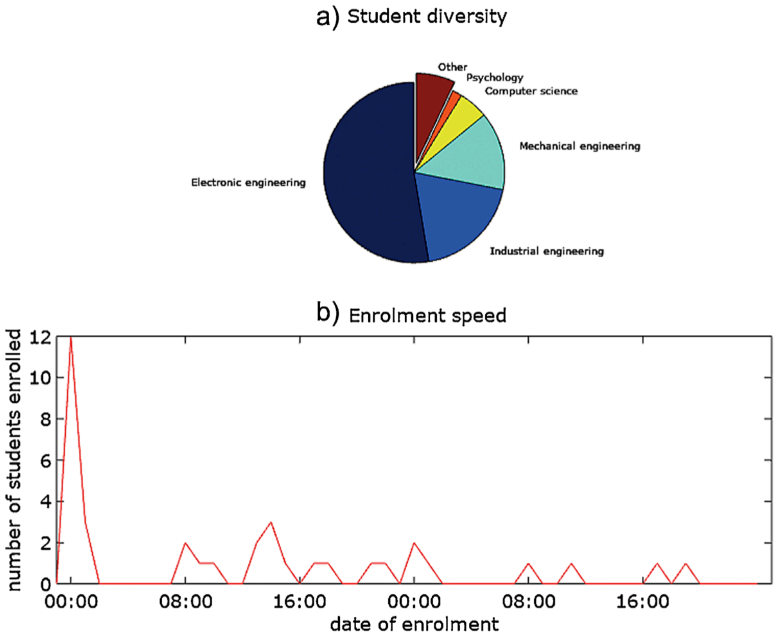


FIGURE 4 (a) Student diversity: the degree and background of our students. Although most of them came from engineering degrees, about 10% came from non-technical degrees. (b) Enrolment for Android programming intensive course: this figure shows how most students enrolled for the 4-edition course in the very first minutes, right after inscription was open at midnight (00:00 hr)

- Version 1. The only functionality of the app is to create new books for the database, which are objects of a class with three members: title, author and rating. Java classes and objects are the key concepts.
- Version 2. The app now stores all books created in an *ArrayList*, and offers functionalities to sort the books by title, author, or rating. Java concepts addressed are collections and sorting.
- Version 3. The app shows improvements in design, by extending the Android *LinearLayout* class. Java concepts addressed: inheritance.
- Version 4. The app now allows us to delete books from the database. The key concept is working with collections in Java.
- Version 5. A new functionality is added: now it is possible to store/load the database from the device's permanent memory. The concept addressed is serialization in Java. Figure 3c shows a screenshot of the app in its final version.

6 | DISCUSSION AND RESULTS

The courses based on Android programming were extremely successful among students, starting with the enrolment

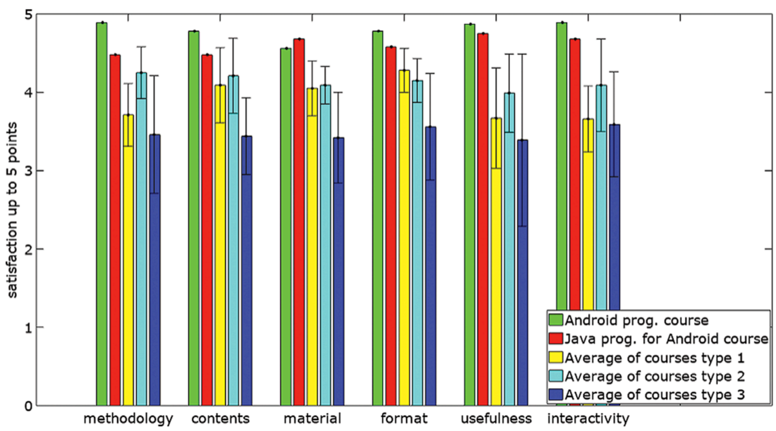


FIGURE 5 Student satisfaction: this graph shows a comparative among the traditional engineering courses where coding is a fundamental competence and the new courses based on Android device programming. In general, most of the students were highly satisfied with both mobile programming courses, slightly preferring the Android introductory course

figures: there were a huge number of applicants for both courses, much more than for traditional Java courses where only console apps are developed. Besides, in the next section we emphasize specific aspects.

6.1 | Student diversity

Figure 4a shows the degree and background of our students. Although most of them came from engineering degrees, about 10% came from non-technical degrees. Besides, not all attendees were young students; there were also applicants from a variety of professions who wanted to update their skills.

6.2 | Interest shown for the courses

We began to offer to the students both Android programming intensive courses in 2012, and at the current time we have carried out 5 editions. Figure 4b shows how most students enrolled for the 4-edition course in the very first minutes, right after inscription was open at midnight (00:00 hours). This reflects the high interest of the students in getting a place on the course, as the maximum number of students was limited to 36. As for the first course, all places were covered extremely fast, in less than 48 hr. The second course also had most places covered in the first week.

6.3 | Student satisfaction

We measured student satisfaction through course surveys. Figure 5 graphically shows the results obtained for the different programming subjects according to the students' assessment of each of the most relevant questions (the same ones indicated in Table 1). Traditional programming courses have been classified into three types, as already indicated in section 4 (type 1: only computer, type 2: electronic equipment or robots, type 3: remote or virtual laboratories), and the mean and the standard deviation of each group of courses is shown. For Android mobile programming courses, the results of each of the courses are displayed individually. In green, the basic course of Android and in red, the most advanced course where the requirements of the Java language are considerably expanded.

In the six selected questions, students clearly demonstrate their preference for mobile device programming courses compared to classical programming subjects. These results confirm the success of the courses, and also allowed us to receive useful suggestions, which were mainly focused on creating new courses for advanced Android and Java programming.

It should be noted that students also value type 2 courses more positively, that is those in which programming is done using specific equipment: motors, microcontroller boards, PLC's, and robots. The lowest results are obtained by courses that use resources based on remote and virtual laboratories.

6.4 | Student engagement

Another important result to measure is student engagement; or students that, after our courses, show interest in Java programming or programming in general. In our case, several students, after attending the Android courses, have chosen app development as topic for their final projects. Some examples of this type of final degree projects can be found in [1,11,16,20].

7 | FURTHER EVENTS

The success of our experiences with Android and Java courses led us to arrange different events in order to increase student interest in programming.

The first event was a hackathon, which was arranged in cooperation with *Supertruper* company, developers of the *Supertruper* app, a price comparator for Spanish supermarkets. The goal of the hackathon was to develop, in only

TABLE 2 Main advantanges of the use of mobile devices in the Engineering programming courses compared to other laboratory equipments

	Robots, development boards, and other electronical equipment	Remote labs	Virtual labs	Mobile devices
Cost	Expensive	Medium	Affordable	Affordable
Student motivation	High	Low	Low	High
Versatility	No	No	No	Yes
1 device per student	No	No	Yes	Yes
Concurrence	No	No	Yes	Yes
Student can learn at home	No	Yes	Yes	Yes
Promotes business abilities	No	No	No	Yes

9 hr, a game that could be added to the app, so that the children can play while their parents do the shopping. A total of nine teams participated, among them students from our courses. More details about this event can be found in the web of the event (http://lcsi.umh.es/hackathon/).

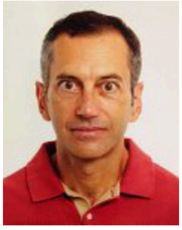
The second event was called *Android Party*. It was an informal meeting where the speakers were university lecturers, professional developers, and also students. Presentations were a mixture of tutorials and demonstrations of developments. The success in terms of attendance resulted in a second edition for the event.

8 | CONCLUSIONS

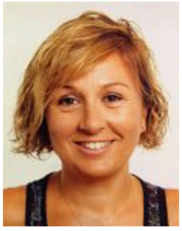
As a conclusion, the use of mobile smartphones and tablets in Java learning courses is extremely useful to motivate students, and can awake their interest in programming. It is no longer necessary to equip the classrooms with robots or other hardware; students carry with them the perfect device to perform software experiments: their own smartphones or tablets. Of all those attending the courses (approximately 350, from 2012), only 3% did not have own Android devices. In Table 2 we summarize the main advantages of the use of mobile devices as a programming tool in the classroom versus traditional laboratory equipment. Nowadays all the students have a smartphone, and in the context of Spanish university, Android is the OS chosen for the majority. So, using the Android smartphone as laboratory equipment is the most affordable electronic device for the university budget. The students can learn to develop apps at the university lectures, but they can also develop at home, something impossible to achieve with the use of robotic systems. The versatility of the apps, solving any type of human demand, links to business creation and marketing and so promotes such great skills in the Engineering students.

REFERENCES

1. A. Aznar, Development of an IOS app in language SWIFT connection external databases and scrapping, Final Degree Project, Miguel Hernandez University, 2014.
2. B. Balamuralithara, P. C. Woods, *Virtual laboratories in engineering education: The simulation lab and remote lab*, Comput. Appl. Eng. Educ., **17** (2009), 108–118.
3. M. Berenguel, F. Rodríguez, J. C. Moreno, J. L.s Guzmán, R. González, *Tools and methodologies for teaching robotics in computer science & engineering studies*, Comput. Appl. Eng. Educ., **24** (2016), 202–214.
4. EasyPIC User's Guide, MikroElektronika, 2014.
5. A. J. Fernández, A. C. Civila, *Practices of advanced programming: Tradition versus innovation*, Comput. Appl. Eng. Educ., **21** (2010), 237–244.
6. C. Fernández, M. A. Vicente, R. Puerto, *Teaching in control subjects supported by virtual experimental equipment (Enseñanza en asignaturas de control apoyadas en equipos experimentales virtuales)*, Proceedings. of VI Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica, 2004, p. 88.
7. C. Fernández, M. A. Vicente, L. M. Jiménez, *Virtual laboratories for control education: A combined methodology*, Int. J. Eng. Educ., **21** (2005), 1059–1067.
8. L. Gomes, S. Bogosyan, *Current trends in remote laboratories*, IEEE Trans. Industr. Electron., **56** (2009), 4744–4756.
9. M. H. Goadrich, M. P. Rogers, *Smart smartphone development: iOS versus Android*, Proc. 42nd ACM Tech. Symp. Comput. Sci. Educ., 2011, pp. 607–612.
10. D. Grunberg, R. Ellenberg, Y. E. Kim, P. Y. Oh, *From Robonova to HUBO: Platforms for robot dance*, FIRA RoboWorld Congress, Springer, Berlin Heidelberg, 2009, pp. 19–24.
11. S. Irles, Design and development of a GPS tracking Android app, Final Degree Project, Miguel Hernandez University, 2014.
12. L. M. Jiménez, R. Puerto, O. Reinoso, R. P. Ñeco, C. Fernández, *Remote Control Laboratory Using Matlab and Simulink*, Proceedings of IEEE International Symposium on Industrial Electronics, (2007), pp. 2963–2967.
13. L. J. Joseph, Rug Warrior Pro Assembly Guide, AK Peters Ltd, 1999.
14. H. Kang, J. Cho, *Case study on efficient Android programming education using multi Android development tools*, Indian J. Sci. Technol., **8**, (2015).
15. G. Licea, J. R. Juárez, L. G. Martínez, L. Aguilar, *Developing programming tools to reach a deeper understanding of advanced programming concepts*, Comput. Appl. Eng. Educ., **16** (2008), 305–314.
16. D. Marhuenda, Development of an educational kids app on Android operating system, Final Degree Project, Miguel Hernandez University, 2014.
17. F. G. Martin, The Handy Board technical reference, Handy Board Documentation, Vol. **15**, 2000.
18. I. A. Martínez, J. M. Angulo Usategui, *Micro'PIC Trainer: Sistema de desarrollo para diseño de proyectos con PIC*, Revista Española de Electrónica, (1997), 80–85.
19. M. Muratet, E. Delozanne, P. Torguet, F. Viallet, *Serious game and students' learning motivation: effect of context using Prog&Play, Intelligent Tutoring Systems*, 11th International Conference, ITS 2012, pp. 123–128.
20. M. Ortiz, Development of Tourist guide Android app, Final Degree Project, Miguel Hernandez University, 2014.
21. A. Sextos, *A paperless course on structural engineering programming: Investing in educational technology in the times of the Greek financial recession*, Eur. J. Eng. Educ., **39** (2013), 18–30.
22. M. A. Vicente, R. P. Ñeco, C. Fernández, R. Puerto, *Experiences with cooperative and competitive learning techniques in Engineering courses*, Proceedings of International Technology, Education and Development Conference, Valencia, 2010, pp. 7–17.
23. M. A. Vicente, C. Fernández, M. N. Robles, *Methodology developed for PIC microcontroller programming (Metodología desarrollada para prácticas de programación de microcontroladores PIC)*, Proc. of VIII Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica, 2008, p. 64.



C. FERNÁNDEZ is associate professor at Miguel Hernandez University, Systems Engineering and Automation Department, since 1997. He obtained his Industrial Engineering degree from Polytechnical University of Madrid (Spain) and his PhD degree from Miguel Hernandez University in Elche. Currently, he teaches linear circuits theory and Java programming. He has participated in several Spanish and European research projects. His main research topics are machine learning and computer vision. He is co-founder of AppAndAbout, a technological spin-off of Miguel Hernández University.



M. A. VICENTE received her MS degree in Electronics Engineering from University of Valencia in 1999. In 2007 she obtained her PhD degree (extraordinary doctorate award) from Miguel Hernandez University. Since 2000, she is working as associate professor at Miguel Hernandez University, Systems Engineering and Automation Department. Her teaching subjects include linear circuits theory and android programming. She has carried research on innovation in engineering education, computational neuroscience, and computer vision. Currently, her research is focused on mobile health.

M. M. GALOTTO is associate professor at Miguel Hernández University, Physics and Computer Architecture Department, since 1999. In 1997, she obtained a degree in Computer Science from Alicante University. In 2004 she obtained a degree in Market Research Techniques by Miguel Hernández University. At present, she is a PhD student in the Science and Industrial and Environmental Technology program at Huelva University. Her current teaching subjects are computer architecture and programming. Her research is focused on the development of new technologies applied to teaching and also in machine learning techniques to monitor professional fishing activity.

M. MARTÍNEZ-RACH received his MS degree in Computer Science from University of Alicante in 1996. He worked for a multinational French-owned computer company as data warehouse analyst. In 2003, he entered at Miguel Hernandez University where at present he works as associate professor in Physics and Computer Science Department. He obtained his PhD degree (extraordinary doctorate award) from Miguel Hernandez University in 2014. His teaching subjects include Operating Systems and Mobile Programing. His research subjects are related to image and video compression specifically with perceptual coding. Currently his research is focused on the perceptual enhancement of the HEVC standard.

A. POMARES is associate professor at Miguel Hernandez University, Physics and Computer Architecture Department, since 1998. In 1995, he obtained his degree in Computer Engineering from Alicante University. In 2001, he obtained his PhD degree from Miguel Hernández University. His teaching activity includes computer architecture and networks, digital electronics, microcontrollers and programming. His current teaching subjects are computer architecture and programming. His research is focused on machine learning applied to monitoring the activity of fishing vessels. He also carries research on new technologies applied to teaching.

How to cite this article: Fernández C, Vicente MA, Galotto MM, Martinez-Rach M, Pomares A. Improving student engagement on programming using app development with Android devices. *Comput Appl Eng Educ*. 2017;25:659–668. https://doi.org/10.1002/cae.21827