Towards education alternatives to teaching and learning of programming: A course experience using open hardware tools

Jonathan Álvarez Ariza, *member, IEEE*
Uniminuto, Department of Electronic Technology, Bogotá, Colombia
jalvarez@uniminuto.edu

***Abstract*—This full paper proposes an educational methodology to teaching and learning of programming using both open hardware tools and industrial components such as Arduino, Raspberry Pi, three phase motors and variable speed drives. The methodology aims to improve the algorithmic thinking in the students due to the deficiencies detected in it, as well as, it searches that the students learn a second programming language, using hands-on designs and activities that require the components mentioned. In addition, the methodology has been employed into the curriculum of a course of programming known as (*open hardware*) for students of technology in electronics (*n=*22), observing that the students learn programming concepts in a better way and their motivation and engagement towards the software area was increased during the course. To technical level, the methodology uses the programming language Python in order to interact with the different ports, peripherals and utilities of the platforms mentioned, creating a link between these elements and the aspects concerning the software design, e.g., data types, classes, methods or Graphical User Interfaces (GUIs). Finally, the methodology takes into account two important aspects, firstly, the skills developed by the students in the proposal are involved into the BoK (Body of Knowledge) of different engineering areas as embedded systems, control and computer science and, secondly, programming is perhaps one of the most difficult areas in engineering but in turn, it has a ubiquitous nature in the areas mentioned.**

***Index Terms*—Python, Programming Education, Open Hardware Platforms, Design Based Learning (DBL), Cooperative Learning (CL).**

I. Introduction

During the last years the field of the education in engineering has been transformed by the new educational technologies such as *Arduino* [1], *Raspberry Pi* [2], *Orange Pi* [3] *or Beagle Bone* [4]. These platforms have allowed to combine hardware and software components in the construction and implementation of designs. Due to the versatility of the technologies mentioned, these can be used to teach and learn programming languages, e.g., Python, C, Java or Dart, fostering the motivation of the students in the programming area. In addition to these features to technical level, programming has gain a ubiquitous nature in several engineering areas as embedded systems, automatic control, robotics, networks and communications.

Thereby, it is needed to educational level, the generation of proposals that take into account the aforementioned and several aspects concerning programming as follows: Firstly, this area is usually one of most difficult areas in engineering because it requires by the student a set of cognitive and metacognitive abilities [5][6][7], e.g., abstraction, analysis, classification, synthesis and modeling. These elements can result overwhelm for the student, if he/she is aware that has deficiencies in them, which could generate lack of motivation, learning problems or even dropping out of the university[5].

Secondly, programming must take into account the main skills or competences relating to the type of engineering (Electrical, Electronics or Computer Science), in which it is taught. For instance, teaching programming in electronics should involve the use of hardware elements that conform a process or system, e.g., sensors, actuators, displays, Programming Logic Controllers (PLCs), etc., with the software concepts tackled along the curriculum. The learning in context help to the students to elaborate a significant learning that contributes to enhancing their educational process and also it allows that the students approaching to programming, applying the concepts learned in designs or projects in educational and professional contexts.

In this regards, the present paper describes an educational alternative to teach programming, using *the programming language Python*. The methodology uses to educational level some elements of the approaches of Design Based Learning (DBL) and Cooperative Learning (CL) in order to improve the *algorithmic thinking*, as well as, to foster the fellowship and the collaborative work so as to attain the learning outcomes proposed in the course. Concerning the technical level, it was employed the open source platforms *Arduino and Raspberry Pi* with the hardware components indicated, creating *scripts* that interacted with them through packets and utilities of Python as *PyFirmata, Matplotlib, PyMata and PAGE.*

According to the elements mentioned, this article is divided in the following sections: section II exposes the background that structures the proposal. Section III, explain the methodology with the description of some practices and basic designs addressed in the course. Section IV discusses the results and findings of the proposal. Finally, section V establishes the final remarks and conclusions of this work.

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

II. Background

In this section, it discusses the background of the proposal divided in the related works and the educational approaches involved in its design.

A. Related works

Rabai *et al.*[8] discuss the current trends of programming languages in the educational and industrial scenarios in US. One interesting point of this research is that Python has gained relevance as language in the first course of programming, occupying the second place in the ranking with a 23.08% of popularity in the period (2010-2013) in the US universities. Moreover, to industrial level, Python is situated in the fourth place of popular programming languages in accordance with TIOBE index [9] that analyses the opinions of skilled engineers world-wide, courses and third party vendors.

Marowka [10] proposes an in-depth educational study for parallel programming that uses Python. According to this research, Python is suitable for teaching Multithreaded programming in novice programmers because offers packages for implementing and debugging different algorithms. However, the author also indicates that Python has current issues in parallel programming but they could be solved if they are used some packages for Message Passing Interface (MPI) as *mpi4py*.

Huei [11] uses Python as language for introduction to programming through robotics. The author argues that Python is an easy language to learn for novice programmers. The methodology proposed was developed in seven weeks, in which the different concepts in programming as variables, operators precedence, mathematical functions, strings and dictionaries were taught to 250 students. The research concludes that the integration of robotics and programming is an effective way to the students create high learning values.

Similarly, Báez *et al.*[12] establish an educational methodology, using a soccer robotic platform. The authors pose a series of learning modules in which integrate concepts of programming and artificial intelligence. The methodology was applied to a group of 30 students of electrical engineering. Within the results described, it shows an improvement of the course grades of an average to 70% to an average of the 82%, checking the pertinence of the proposal.

On the other hand, Rao *et al.*[13] employed a methodology in an embedded system course that gathered IoT devices, Python and Raspberry Pi. The course enrolled 8 students of Master degree that proved the methodology, modifying the curriculum that consisted in to teach the architecture of ARM processors. The study indicates that the platform Raspberry Pi provides an easy way to experiment in the embedded system area that can enhance the educational process of the students.

Finally, the investigation carried out by Almeida *et al.*[14] shows how the remote laboratories can be employ to teach programming through NXT Lego robots and Python. The participants of this research indicated that using robotics and remote laboratories provided them an easy way to learn programming, in fact, the 71% of the users fully agreed, and 29% partially agreed with this premise.

All investigations mentioned are relating to this work, in the sense, to offer a perspective of use of Python in different scenarios such as robotics, parallel programming, IoT and embedded systems. Despite these important researches and studies, there is a current lack of proposals that integrate the teaching of programming and the industrial field. In this way, the present research contributes to strengthen the current state of art in the educational proposals that take into account this issue.

B. Educational approaches

1) Design-Based Learning (DBL): Puente *et al.*[15] have defined the concept of Design-Based Learning (DBL) as an educational approach focused on the inquiry and reasoning processes with the aim to produce innovative objects, e.g., artifacts, systems, etc. From pedagogical paradigm, the DBL takes in mind both the problem solving theory and the practices in the development of projects focused on the learner. From our perspective, we understand the DBL as an educational methodology based on cognitive and metacognitive processes, e.g., (*conceptualization, analysis, design, implementation and debugging*) that are important into education in programming and the design of any algorithm.

Hence, one important element into the structure of the DBL approach is the *problem*. Regarding to this, Jonassen[16] indicates that the problem is an unknown entity in a context, e.g., social, technical or cultural. The transition between the problem and its solution requires two elements: On one hand, a mental model of the problem and by the other hand, the manipulation mental or physical of it[16].

Furthermore, the DBL approach uses the *design* as source of knowledge[17], this is, through the design, the students perform actions such as planning, modeling, abstraction and analysis, so as to create solutions that respond to certain requirements. The main aspects related to the DBL approach are summarized as follows[17]:

- In relation to the design, the problem is explored with its implications into the practice. Afterwards, it should develop a design methodology that tackles these implications in the educational field.
- The assessment has a constructivist nature in the educational process of the students.
- In the social context, *the collaborative or Cooperative Learning (CL)* is an important methodology that helps to achieve the goals of a determined design through the interaction and learning between classmates.
- The designs or projects worked with the students should be authentic, innovative and multidisciplinary.
- The teacher leads the process of the design which is based on tasks.
- The learning is a process given mainly by hands-on activities.

2) Cooperative Learning (CL): As it mentioned, the CL is an important methodology that articulates the DBL approach and thus, the methodology proposed. Johnson [18][19] defines the CL as a methodology that operates in reduced groups in which the learners work together with the aim to maximize the learning both a personal level as group level. In this way, in this methodology exists five essential elements which are sorted below:

- *Positive interdependence:* This term is refereed to how the students work together and they assume an engagement both with their own and collective learning. For the members of a group is clear that in order to achieve the goals proposed (technical and educational), it must exist the gathering of individual and collective efforts.
- *Individual and group responsibility:* Each member of the group must understand that has an individual and group responsibility in the different tasks assigned. The awareness of this responsibility guarantees the fulfillment of the purposes proposed in a project or design.
- *Stimulating interaction:* The learners should work together in tasks that foster the successful in the overall goals such technical as educational in a determined project or design. This fact stimulates the collaborative learning of each member in the group.
- *Interpersonal and group practices:* Not only are needed the contents of a subject, also the cooperative learning must promote the fellowship, the teamwork, the communication and the solution of conflicts presented in the group.
- *Group assessment:* The members of the group must analyze their degree of progress in the learning and if their goals about this have been achieved.

Moreover, another important component in the CL methodology is the kind of groups that could be conformed. Katzenbach *et al.*[20] have classified these groups in four categories: pseudo-group, traditional group, cooperative group and high performance cooperative group. In a pseudo-group the students work independently without an integration between them. In a traditional group the students work in group but the tasks proposed in the course are not structured for a truthful teamwork while in a cooperative groups the students work together with good disposition in concordance with the learning goals defined. In the case of high performance cooperative learning groups, the positive interdependence overcome any expectation in terms of performance such personal as collective.

These approaches helped to conform from pedagogical reference the proposal that concern to this work.

III. Methodology

This section exposes the main components of the methodology from the educational and technical levels.

A. Educational component

In function of the background mentioned, this section discusses the educational aspects relating to the proposal.

The proposal tackles three main aspects: Firstly, there are some problems in the algorithmic thinking detected in the students, that limit the learning of the concepts in programming and the way in which them construct an algorithm. Secondly, the need to put the programming in context, as part of an integral formation that takes into account the skills and competences that must be developed by the students in electronics engineering or technology. Thirdly, it searches to increase the motivation to learn programming, drawing on the educational and technical features of some open hardware platforms as Arduino and Raspberry Pi [21][22][23]. These elements have allowed to structure the methodology as is depicted in the Fig.(1).

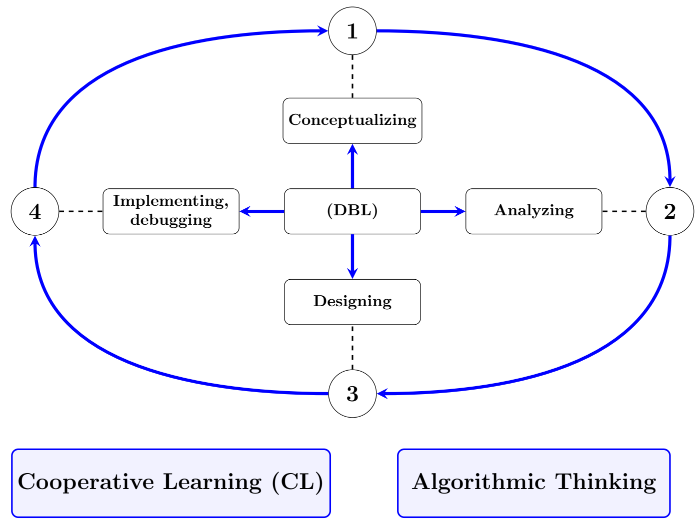

Fig. 1. Educational scheme of the methodology with its main moments.

In this scheme, there are four moments in the sequence of learning of a student: *Conceptualizing, analyzing, designing, implementing and debugging*, that are articulated with the DBL approach. As basis or pillars, it has integrated the CL methodology and the Algorithmic Thinking that were developed through the course. The structure of the course was addressed in a regular semester in 16 academic weeks with an intensity of two hours per week.

Concerning the contents, these are illustrated in the Table.(I), respectively. The first part of the course explored the concepts of programming with Python and its features. Afterwards, the concepts learned were employed to interact with Arduino platform, Python and some packages for this function as *Pyfirmata and PyMata*[24][25]. In the second part of the course, the students took the contents associated to the Raspberry Pi platform, starting with the configuration of Linux Raspbian so as to access remotely or directly to it and subsequently, employing sensors and actuators as (servos, three-phase motors and speed variable drives), handling them through Python. These elements will be used in the future labor life of the students.

During the methodology were taken two moments for the *assessment*, where it was collected the students' opinions with the aim to improve the educational aspects of the methodology. We consider that the assessment should be formative and it have a constructive nature. Furthermore, any error presented in the algorithm of the student, it was corrected in each practice, making that the student understood the problem and corrected it.

TABLE I
COURSE ORGANIZATION AND STRUCTURE OF TOPICS BY WEEKS.

Week	Topics
1-2	Fundamentals of Python (features of the language, data types (tuples, lists, dictionaries, arrays), loops, packages and their installation).
3	Concepts relating to the platform Arduino, and first programs.
4	Firmata Protocol in Python. Reading and writing digital I/O ports with the *Pyfirmata* package.
5-6	Sensors and actuators (temperature, resistance, humidity, three phase motors) with scripts in Python using *PyMata and Pyfirmata* packages.
7-8	Design of GUIs through Tkinter and PAGE in Python that interact with Arduino, sensors and actuators.
9	Assessment and feedback of the methodology in regards to the learning in programming with Arduino, the contents and the course motivation.
10	Fundamentals of Python with Raspberry Pi. Introduction to Linux Raspbian, configuration of General Purpose I/O (GPIOs).
11-12	Remote access to Raspberry Pi board through SSL protocol. Useful commands in console of Linux Raspbian to execute Python scripts.
13	Configuration of sensors and protocols as I2C or SPI in Python with Raspberry Pi.
14-15	Design of GUIs through Tkinter and PAGE.
16	Assessment and feedback of the methodology in regards to the learning in programming with the Raspberry Pi board, the contents and the course motivation.

In each session per week, it was proposed *small designs* that addressed the contents mentioned and the moments described in the Fig.(1), reinforcing the Algorithmic Thinking and the positive interdependence that conforms the core of the CL approach. Some of these designs will be described in the next section.

In order to interact with Arduino, it was proposed a scheme that utilizes the packages Pyfirmata and PyMata as is depicted in the Fig.(2). In 1, the students employed the *Pyfirmata* protocol which is a Python package that allows to communicate with the Arduino boards through the *Firmata protocol* [26] that implements actions as reading and writing of digital pins and writing the duty cycle for the PWM pins.

As for the package PyMata, it allows the implementation of the I^2C protocol in Python for the Arduino boards with the Firmata protocol. In 2, the communication is made through serial port physical or virtual. For the methodology, the students worked with the board *Arduino Mega 2560* whose main features are: Processor (ATmega 2550) running at frequency of 16MHz, 54 I/O pins, 16 analog pins, 15 PWM outputs, Flash Memory of 256KB, UART, I^2C and SPI peripherals.

In regard to Raspberry Pi platform, it was utilized the *version 3 model B* with the following features: CPU ARM-Cortex A53 running at 1.2GHz, 1GB of RAM, Bluetooth, Ethernet and Wi-Fi modules. Due to Linux Raspbian contains a distribution of Python (version 2.7), the programs were elaborated directly in it. So as to interact with the General Purpose I/Os (GPIOs) and the peripherals as I^2C of the Raspberry Pi, the students installed and used some packages as RPi, time and SMBus. The scheme proposed for Raspberry Pi and Python is shown in the Fig.(3).

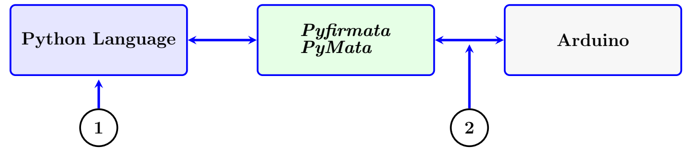

Fig. 2. Scheme proposed for Arduino boards and Python through packages Pyfirma and PyMata.

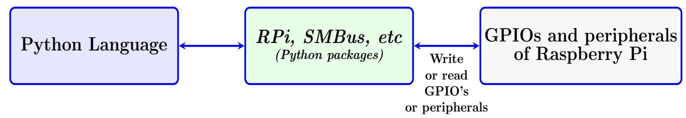

Fig. 3. Scheme proposed for Raspberry Pi and Python.

B. Description of some designs

This section describes some designs that the students developed in function of the aspects educational and technical mentioned. Our intention is to provide some examples of the *designs* made by the students with Python and the platforms indicated.

1) Design 1: Starting up of AC Induction Motor (ACIM) with Arduino: The aim of this design was to introduce to the student to use the package *Pyfirmata* with its particularities, as well as, the student learned some loops and statements in Python, creating an algorithm to start up an ACIM. The students also used some industrial components as relays, contactors, breakers and they calculated the current and voltage parameters in order to implement them in the design. The Fig.(4) depicts an example of the design made by the students. Through simple buttons, the students commanded the start up and stop of the ACIM. An example of the program in Python is shown in the Algorithm (1).

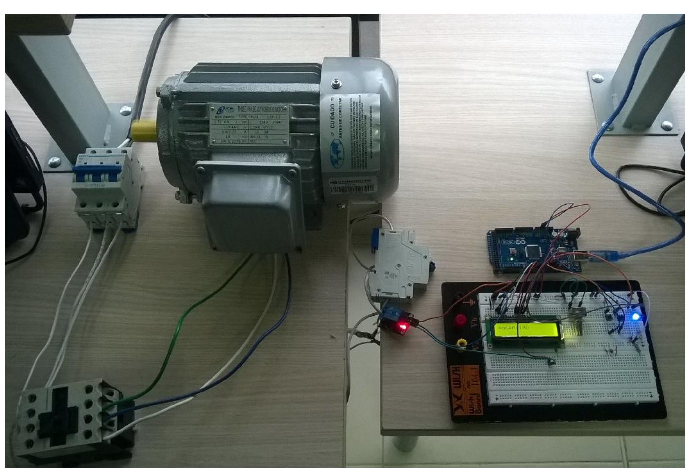

Fig. 4. *Design 1*. The figure shows the components used by the students.

Algorithm 1 Example of code for *the design 1*.

```
import pyfirmata  # import pyfirmata package
# import ArduinoMega class
from pyfirmata import ArduinoMega
import time #import time package
#Arduino's working port
board= ArduinoMega("COM13")
#Declare iterator prevents buffer overflow
it = pyfirmata.util.Iterator(board)
it.start()  # start iterator
#Push button on Pin 4 (digital) of ArduinoMega
pin = board.get_pin('d:4:i')
while True:#repeat forever
        value = pin.read()  #Read pin's state
        #if push button was pressed
        if value is False:
                #start motor
                board.digital[13].write(1)
                #Print in console
                print ("Button pressed")
        time.sleep(0.1)#delay of 0.1 seconds
```

2) Design 2: Emergency stop for over temperature in ACIM through board Raspberry Pi: In this design, it was worked with the package *matplotlib* in so as to plot the temperature sensed by the device TMP102. This sensor has an I^2C protocol and a 13 bit resolution for the temperature in a range between -40°C and 125°C. The aim of the design was, on one hand, that the students learned the configuration of the I^2C protocol through Python and some bit operations, and by the other hand, that the students employed the package matplotlib to plot data sensed. The bit operations were used to transform the *data stream* read from the sensor to the equivalent in temperature.

Moreover, the students designed the script in Python and implemented it, to generate an over temperature control for the ACIM. The Algorithm (3) describes some parts of the example of code made by the students and the Fig.(5) shows the design implemented.

Fig. 5. *Design 2*. The students designed an emergency stop for the ACIM and they plotted the temperature sensed by the device TMP102 through Matplotlib in Python.

3) Design 3: Speed control of the ACIM from Arduino: The purpose of this design was to control the speed of the ACIM, employing variable speed drives as the Telemecanique altivar 31, Optidrive E1 or Sinamics G110. The students modeled the voltage over the digital potenciometer XC9C103 in function of the ACIM's speed and they configured the variable speed drive with an analog reference. The ACIM's speed was controlled using the console of the IDE (PyCharm community edition).

Algorithm 2 Example of code for *the design* 2.

```
import smbus #import I2C package
#use plot from matplotlib
import matplotlib.pyplot as plot
# import time package
import time
#0 = /dev/i2c-0
#1 = /dev/i2c-1
plot.ion()#start graphic
analog=[] #Define array that will contain the
    data
I2C_BUS = 1
#...
def Reading():
        temp_reg_12bit = bus.read_word_data(
            DEVICE_ADDRESS , 0 )
        temp_low = (temp_reg_12bit & 0xff00)
            >> 8
        temp_high = (temp_reg_12bit & 0x00ff)
        #Convert the value read to the
            equivalent in temperature
        temp  = ((( temp_high * 256 ) +
            temp_low) >> 4 )
        temp_C = float(temp) * 0.0625
        #...

def Plotting():
        plot.hold
        plot.ylim(0,150)  #Put Y axis limits
        #...

while True:
        Reading()
        drawnow(Plotting) #Update the plot
        time.sleep(0.1)#delay of 0.1 seconds
```

Fig. 6. *Design* 3. The students configured the parameters of the variable speed drive Sinamics G110. The speed was handled through a digital potentiometer in the Arduino Mega 2560.

IV. Results and Findings

In this section, we discuss the results and findings of the methodology. It is important to mention that the methodology was applied to 22 students ($n = 22$) of the program of Electronics Technology during the first semester of 2018 inscribed into the course of *Open Hardware*. In the first part of the course, it was elaborated a survey with the purpose to know the perception of the students about the concepts and knowledge that they had in the subject of *basic programming* and also the different difficulties in it. This subject is offered to our students and it is focused on the programming language C. The conclusions of this survey are summed up as follows:

1) 91.6% of the students consider that if it had articulated the use of hardware devices in the structure of the course of programming, their learning would have been much more *meaningful*. Moreover, 100% of the students argued that the course needs a better articulation with hardware components and platforms.
2) One question of the survey consisted in to evaluate in a scale to 1 to 5, where 1 is a lowest value and 5 is the highest value concerning the learning of the students in the subject of basic programming, the mean of the grades in this process was 3.33.
3) With regard to the problems in programming, the students indicated that they have difficulties in: Search errors in programming and fixing them; order in the sequence of code (algorithmic thinking); library creation; declaration of variables; use of conditionals and statements.

These elements were taken in the structure of the methodology. In the last part of the course was made a survey according to Likert choices to assess the methodology and to compare the skills and knowledge in the students. The results of this survey are synthesized in the following tables and figures.

TABLE II
Survey of assessment of the methodology (Part I).

Question \ answer option	**Completely**	**Quit a bit**	**A little**	**None**
1. Do you consider that the methodology contributed to your learning in programming?	45.46% (10)	54.54% (12)	0% (0)	0% (0)
2. Did the practices allow me to put in a real context the algorithms designed in Python?	45.46% (10)	54.54% (12)	0% (0)	0% (0)
3. Has the methodology allowed me the cooperative learning, that is to say, interact, share and learn with other classmates?	54.54% (12)	45.46% (10)	0% (0)	0% (0)
4. Has the methodology allowed me to get interested in programming?	86.36% (19)	13.64% (3)	0% (0)	0% (0)

TABLE III
Survey of assessment of the methodology (Part II).

Question \ answer option	**Yes**	**No**
5. With the employment of the methodology, Have you articulate in an easy way hardware components as sensors or actuators with Python Scripts?	100% (22)	0% (0)
6. Do you consider that Python is a powerful programming language in comparison with other languages that you have used as C, Java, etc.?	100% (22)	0% (0)
7. Has the methodology helped me to understand in a better way the programming algorithms?	100% (22)	0% (0)

The survey searched the point of view of the students regarding the knowledge of programming, the algorithmic thinking developed, the real context of the practices, the designs elaborated and the motivation towards programming. The methodology was evaluated in a scale of 1 to 10 by the students, the mean of this assessment was 9.3181, which demonstrates that the proposal had a good acceptance between the students.

In the same way, it were compared the means of the students' grades in the three terms of the semester (scale 0 to 5) without and with the methodology in the (periods 2017-II and 2018-I) respectively. This comparison is shown in the Fig.(7).

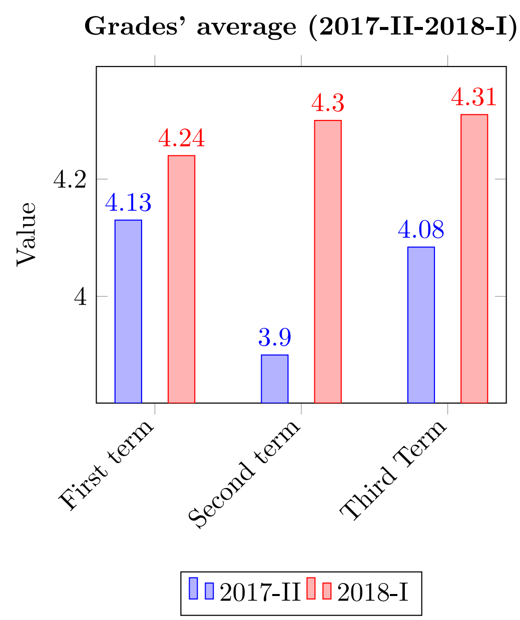

Fig. 7. Comparison of the students' average grades in the periods (2017-II, 2018-I) by academic terms.

A. Discussion

According to the outcomes presented, firstly, the methodology attained an increment of the grades in the course because the motivation to learn programming was fostered as is depicted in the table II and Fig.(7). The methodology was provided to students of technology in electronics. As contextualization, in Colombia, the technological programs are understood as a kind of programs focused on labor fields, with a duration between two or three years. Most of the subjects established in the curriculum search that student develops the abilities that a determined sector needs, e.g., in some programs of electronics technology their emphasis include telecommunications, industrial communications, industrial automation or bioengineering [27].

Secondly, the proposal provided to the students different programming skills that can be used in the industrial sector with different types of devices such as actuators, variable speed drives or PLCs. The proposal also strengthened the algorithmic thinking in the students through hands-on designs and activities as it was indicated by the following students:

Student A*: Through the methodology, I implemented functional programs with electronic components that can be handled to industrial level.*

Student B*: The hands-on laboratories made that the learning was autonomous. One interesting element was the teacher's support to correct the different programming mistakes.*

Student C*: I consider that the teaching technique allowed me understand the principles of programming that I wanted to reinforce.*

When programming goes joined with practical activities in context, it can be achieved a better motivation to learn, as well as, an improvement of the algorithmic thinking in the students.

Thirdly, the students considered that Python is a practical language because they wasted less time in programming and debugging in comparison with other languages. It is an important feature of the high level languages with interpreter. Some comments concerning this are shown as follows:

Student D*: Python is interesting because it allows to interact easily with hardware components.*

Student E*: I consider that the structure and syntax of the Python language is easier to learn if you compare with other programming languages.*

Fourthly, although Cooperative Learning (CL) is difficult to achieve in practice, the methodology searched that the students worked in group and shared their experiences so as to attain the learning goals. Finally, the educational platforms employed in the methodology (*Arduino and Raspberry Pi*) have allowed different types of designs and their implementation by the students in an easy way as it was mentioned several times by them. This result is aligned to some references of the related works described.

As difficulties, it is important to mention that the time dedicated to programming in the methodology is reduced. The students indicated that should exist much more time for programming and the designs assigned, which would allow the learning of other concepts involved in programmming and Python.

V. Conclusions and Future Work

In this paper, it has been presented an educational methodology that involves programming in Python and hardware components in the industrial field in order to improve the algorithmic thinking and the skills of the students in the software area. It has used these elements due to the course is provided to students of technology in electronics and they must develop certain abilities and skills between them, the suitable articulation of hardware and software in designs, projects or applications. Moreover, it was employed the DBL and CL approaches so as to enhance the educational process of the students, particularly, the component of the algorithmic thinking. The students shared, interacted and learned in group according to the practices and designs proposed in the different sessions of the course, therefore, programming was grounded to a real context in function of the educational needs of the students. As future work, on one hand, the methodology should be validate in other subjects of the curriculum, observing the pertinence of it. By the other hand, to technical level, we want to integrate other hardware components mainly focused on Internet of Things (IoT) with the industrial field and programming in Python.

Acknowledgment

The author wishes to acknowledge to Uniminuto University for the financial support of this project. The author also wishes to extend their acknowledge to the academic community of the Electronic Technology program specially to the students of the open hardware workshop and the automatic control research incubator. For all them, thanks for their help. Finally, the author acknowledges specially to the reviewers for their important contributions in the structure of this work.

References

[1] (2017) Introduction to arduino. [Online]. Available: https://www.arduino.cc/en/Guide/Introduction

[2] R. P. Foundation. (2017) Raspberry pi specifications. [Online]. Available: https://www.raspberrypi.org/

[3] O. Pi. (2018) Orange pi specifications. [Online]. Available: http://www.orangepi.org/

[4] B. Board. (2018) Beagle bone specifications. [Online]. Available: https://beagleboard.org/bone

[5] D. Teague, "Pedagogy of introductory computer programming: A people-first approach," Master's thesis, Queensland University of Technology, 2011.

[6] J. Á. Ariza, "A proposal for teaching programming languages through open hardware tools," in *2016 IEEE 8th International Conference on Engineering Education (ICEED)*, Dec 2016, pp. 202–207.

[7] A. Kak. (2016) Teaching programming. [Online]. Available: https://engineering.purdue.edu/kak/TeachingProgramming.pdf

[8] L. B. A. Rabai, B. Cohen, and A. Mili, "Programming language use in us academia and industry," *Informatics in Education*, vol. 14, no. 2, p. 143, 2015.

[9] T. software company. (2018) Tiobe index. [Online]. Available: https://www.tiobe.com/tiobe-index//

[10] A. Marowka, "On parallel software engineering education using python," *Education and Information Technologies*, vol. 23, no. 1, pp. 357–372, 2018.

[11] Y. C. Huei, "Benefits and introduction to python programming for freshmore students using inexpensive robots," in *2014 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, Dec 2014, pp. 12–17.

[12] H. Báez, K. Pérez, E. Rojas, S. Rodríguez, J. López, C. Quintero, and J. M. Calderón, "Application of an educational strategy based on a soccer robotic platform," in *2013 16th International Conference on Advanced Robotics (ICAR)*, Nov 2013, pp. 1–6.

[13] A. R. Rao, D. Clarke, M. Bhdiyadra, and S. Phadke, "Development of an embedded system course to teach the internet-of-things," in *2018 IEEE Integrated STEM Education Conference (ISEC)*, March 2018, pp. 154–160.

[14] T. O. Almeida, J. F. d. M. Netto, and M. L. Rios, "Remote robotics laboratory as support to teaching programming," in *2017 IEEE Frontiers in Education Conference (FIE)*, Oct 2017, pp. 1–6.

[15] M. Gomez Puente, M. van Eijck, and J. W., "A sampled literature review of design-based learning approaches: A search for key characteristics," *International Journal of Technology and Design Education*, vol. 23(3), pp. 717–732, 2013.

[16] D. H. Jonassen, "Toward a design theory of problem solving," *Educational Technology, Research and Development*, vol. 48 (4), pp. 63–85, 2000.

[17] S. M. Gómez Puente, *Design based learning: exploring an educational approach for engineering education Eindhoven*. Technische Universiteit Eindhoven, 2014.

[18] D. W. Johnson and R. Johnson, *El aprendizaje cooperativo en el aula*, P. SAICF, Ed., 1999.

[19] D. W. Johnson, R. T. Johnson, and S. K. A., "Cooperative learning: Improving university instruction by basing practice on validated theory," *Journal on Excellence in University Teaching*, 2013.

[20] D. Katzenbach, J Smith, *The wisdom of teams*. Harvard Business School Press, 1993.

[21] A. Garrigós, D. Marroquí, J. M. Blanes, R. Gutiérrez, I. Blanquer, and M. Cantó, "Designing arduino electronic shields: Experiences from secondary and university courses," in *2017 IEEE Global Engineering Education Conference (EDUCON)*, April 2017, pp. 934–937.

[22] P. Jamieson and J. Herdtner, "More missing the boat - arduino, raspberry pi, and small prototyping boards and engineering education needs them," in *2015 IEEE Frontiers in Education Conference (FIE)*, Oct 2015, pp. 1–6.

[23] X. Zhong and Y. Liang, "Raspberry pi: An effective vehicle in teaching the internet of things in computer science and engineering," *Electronics*, vol. 5, no. 3, 2016. [Online]. Available: http://www.mdpi.com/2079-9292/5/3/56

[24] (2018) Pymata package specifications. [Online]. Available: https://github.com/MrYsLab/PyMata

[25] (2018) Pyfirmata package specifications. [Online]. Available: https://github.com/tino/pyFirmata

[26] (2018) Firmata protocol specifications. [Online]. Available: https://vvvv.org/documentation/firmata-protocol

[27] C. A. Herazo and J. Á. Ariza, "A proposal of educational model for research incubators in technological programs of electronics," in *2016 IEEE 8th International Conference on Engineering Education (ICEED)*, Dec 2016, pp. 143–148.