ABSTRACT Developing programming competencies is essential for systems, information science, computer science, and electrical engineering students. Engineering students usually face the complexity of working with programming languages that demand compliance with syntactic and semantic rules, which typically represent a daunting task for novice students. Watching textual messages on the screen only, like the classic “hello world,” is no longer attractive in the current information society, a missing motivation and possible obstacle to developing programming competencies. Students would like to interact with hardware and appreciate environmental reactions. Arduino board permits developing solutions like that. This article presents the academic experience of first-year students of Ingeniería de Sistemas e Informática at the Universidad Continental (ISI-UC) of Huancayo, Peru, using the Arduino microcontroller board for the teaching-learning process to develop programming competencies. The results obtained show a positive impact regarding the experience of previous using traditional text-based programming languages. Using Arduino, students create digital circuits and computational electronics competencies, another significant benefit. This experience used an online simulator, and the results obtained permit us to plan future online education strategies for this major. The next step will be the application of Arduino and the online simulator to deepen programming skills, including recursivity, real-time constraints, multitasking features, data structure, data-oriented programming, and object-oriented programming. The primary limitations encountered in this experiment were the students’ lack of experience with electronics concepts to build circuits and, in some cases, the low internet speeds to assist in the programming process of online education. Realizing simulated experiences in classroom experiences was not a significant challenge for teachers and most students. However, problems could arise when students perform practical experiences using real Arduino boards in traditional classes for the availability requirement of Arduino and other electronic components.

INDEX TERMS Programming competencies, Arduino, algorithms, online education, electronics.

I. INTRODUCTION

In computer science, programming or writing a program means giving a set of instructions to a computer or microcontroller for its execution and then achieving a definite purpose; programs written in programming languages are called algorithms [1]. According to Vidal et al. [1], an algorithm represents a set of precise, finite, and unequivocal steps to obtain a defined goal. Programming can be a pleasant experience like writing poetry or composing music. Stimulating the development of algorithmic thinking in young students represents the established objectives of educational systems in Europe [2], [3] and South America [4] because the development of algorithmic competencies enables the development of analysis, design, and problem-solving skills [1]. However, as stated by [5], developing these competencies turns out to be a teaching-learning problem for both teachers and students.

The development of algorithmic and programming languages competencies is essential in the training of professionals and technicians in systems and information sciences [6]. Introductory courses to algorithms and programming are common in the first semesters of such careers. For example,
the curriculum of ISI-UC in Huancayo - Perú [7] incorporates the Fundamentals of Programming (FoP) course in both academic semesters. Likewise, the new curriculum of Ingeniería en Información y Control de Gestión at the Universidad Católica del Norte - Chile (IICG-UCN) includes a programming course during the first-semester [8], which is repeated in the second semester as a replay course. We can observe a similar situation at the University of Tarapacá, Chile, where programming competencies are required to strengthen other areas of Electrical Engineering.

A. PROBLEM STATEMENT, GOAL AND CONTRIBUTIONS
As Qian and Lehman [9] point out, the science of programming, in addition to problem-solving competencies, considers abstract concepts strongly connected with the mechanics of programming and the practical domain of syntactic and semantic structure of the applied programming languages. Vidal et al. [1] argue that developing programming skills using text-oriented programming languages requires high-level abstraction and mainly algorithmic reasoning skills. However, students who regularly use textual programming languages spend more time understanding and adhering to the syntax rules instead of developing algorithmic and problem-solving competencies. That is a relevant issue nowadays because students aim at learning as fast as possible and are more familiar with visual devices than earlier generations [10]. Hence, teachers and professors need to identify adequate tools for the teaching-learning process for developing programming competencies [6].

Current technology has developed robust and non-expensive electronic programmable devices such as micro-controllers. The massive academic and industrial use of Arduino has led to many shared open-source libraries and tools for the entire community. In that field, the Arduino board represents a very affordable and comfortable environment that invites both students and educators to develop several applications for their everyday life situations [11]. For example, in electrical engineering, the work of Guerrero-Rodriguez et al. [12] show the application of Arduino for developing a low-cost circuit emulator and applying digital filter algorithms. Likewise, the work of Jeong et al. [13] describe the successful use of Arduino for emulating a low-cost radar system with signal processing for the object detection algorithms. Applying an open-hardware and open-software philosophy, the main goal of this work is to answer the following research questions.

RQ1 [Impact of Arduino in Programming] How have Arduino affected the development of programming competencies at the ISI-UC students?

RQ2 [Students’ Perception] How did students at the ISI-UC perceive the benefits of using Arduino for developing programming competencies?

RQ3 [Motivation to Continue Using Arduino] How motivated are you to continue studying and using Arduino in the future?

This article describes academic experiences using the Arduino system to develop programming competencies in first-year university students of a programming course in information science, computer engineering, and electrical engineering majors. Specifically with ISI-UC students. We can find material regarding the ISI-UC course, homework, and students score available in a GitHub repository.1 Due to the high level of interaction with the environment achieved when using Arduino systems, the results obtained demonstrate that the development of algorithmic and programming skills is highly feasible, along with the development of basic electrical and electronic knowledge. Hence, the followings are the main contributions of this paper:

• First, to highlight the use of Arduino for developing algorithmic-thinking and programming competencies in students
• Second, to show the usefulness of Arduino in developing electronic theory and practical knowledge
• Third, for developing competencies in implementing systems for digital signal acquisition and processing

This article extends the work of Tupac-Yupanqui et al. [14]. The rest of this paper is organized as follows. Section II describes and exemplifies the use of Arduino. Section III details the competencies and experiments in the FoP course of ISI-UC applied in 2020. After that, in Section IV, the academic results obtained by using Arduino as a teaching-learning methodology for introductory programming courses are discussed. Section V presents similar studies regarding the benefits of using Arduino to develop academic competencies. Section VI details a few practical issues. The paper concludes by summarizing the benefits of our academic experience and detailing the motivation for continuing with it in the current and future years.

II. ARDUINO BOARD BACKGROUND
We provide details of the open-hardware Arduino platform and the Tinkercad simulator to design and evaluate Arduino systems. Though Tinkercad does not consider all the possible components to integrate into an Arduino system, this platform integrates hardware components capable of implementing real experiments. Figure 1 shows two basic examples of hello world in electronics to connect an LED light to a 9-volt battery and turn it on: the first example shows the case where an excessive amount of electrical current burns the LED out, whereas the second one illustrates the right way to make the connection and turn on the LED adequately. This example enables us to understand practical details about the Ohms law, an empirical principle that shows the proportionality between electrical current and voltage [15].

Thanks to technological advances in both hardware and software, the development of integrated hardware and software solutions for students and professionals in both areas is becoming increasingly feasible. Though desktop computing systems reflect great advances in computing speed

1https://github.com/cvidalmsu/IEEEAccess-Arduino-Programming
The design of electronic solutions seems a complex task due to the physical devices involved in implementing the final electronic circuit. This risk disappears with the use of simulators. Arduino is an open-source electronic platform for developing interactive computing and electronics projects [20]. The Arduino programming language is Wiring [21], whose syntax and semantics are like those used by C and C++ programming languages [22]. Prima facie, electronic design and programming represent complex tasks for novice programming students. The web platform Tinkercad [23] seeks to solve the programming and design issues of Arduino systems thus:

1) Tinkercad unifies a block-based programming language like Scratch [17], [18], [24] and the Wiring programming language.
2) Tinkercad permits using a set of electronic components for integrating the Arduino in the design and program computing solutions.

A. TINKERCAD CIRCUITS

Tinkercad [23] is a free and easy-to-use web platform for 3D design, electronics design, and coding in developing electronic computing solutions using Arduino. Arduino was created in 2005 in Italy, seeking to easily integrate computing and electronics for students in low-cost and open hardware contexts [25]. Today, there is a diverse set of inexpensive Arduino-compatible hardware and software items on the market, such as liquid crystal displays, sensors, and built-in web development platforms like Tinkercad.

Tinkercad supports and encourages the development of solutions with Arduino [25] and includes a design area for the hardware components in the solution, a classification of different hardware components capable of adding, and a code area for writing the program. In the Appendix VII, Figure 5(a) shows a hardware solution designed in Tinkercad to turn on a red LED, where the Arduino board is in charge of sending the positive and ground electrical signals.

Tinkercad permits programming Arduino solutions using a block-based programming language like Scratch [24]. Figure 5(b) in the Appendix VII shows the coloured ranking of the blocks according to their purpose: Output (blue), Input (purple), Notation (grey), Flow Control (yellow), Math (green), and Variables (pink). In the Appendix VII, Figure 5(c) shows the structured blocks and Wiring programming language code to cyclically turn the LED on and off at times t1 and t2, respectively. Wiring programming language uses C and C++ syntax and semantics. Hence, we can observe that programming Arduino requires advanced programming knowledge for novice students. As a solution, Tinkercad can generate the Wiring code for the Arduino solution from a block-based code solution. That is an excellent feature for programming beginners to understand the programming functionalities of Wiring quickly. Thus, Tinkercad is a superb web platform for learning the Arduino-based electronic system to develop simple solutions.
III. METHODOLOGY AND RESEARCH DESIGN
The study program of FoP at the ISI-UC defines the following competencies and learning objectives:

1) Ability to define an algorithmic solution to a computational problem [26].
   1.1 Theoretical/practical command of input/output operations in algorithmic solutions.
   1.2 Theoretical/practical domain of variables in algorithmic solutions.
   1.3 Theoretical/practical command of control structures in algorithmic solutions.
   1.4 Theoretical/practical command of iteration structures in algorithmic solutions.
   1.5 Theoretical/practical domain of subroutines and modular thinking in algorithmic solutions.
2) To write computational solutions in a programming language.
   2.1 Represent algorithmic solutions in a programming language.
   2.2 Measure feasibility and practical efficiency of programming solutions.
3) Ability to work in a team [27].
   3.1 Management of responsibilities associated to team tasks.
   3.2 Self-evaluation and co-evaluation of collaborative work.

Considering the competencies and learning objectives of FoP, the research questions and the main goal of this article, we configured several environmental and instructional conditions in the introductory programming course at the ISI-UC major for their achievement. That course started in both academic semesters during 2020, using Arduino and the Tinkercad platform from the eighth week. During the first seven weeks, we used the PSeInt [28] and Raptor [29] tools for developing the algorithmic thinking. The course was conducted online during 2019, 2020, and 2021 because of the world’s critical health situation preventing face-to-face education.

In the first session of week eight, we started reviewing a background of Arduino using the examples of Figures 1-5. In the following four sessions, we highlight the basic structure of Arduino solutions (setup configuration and loop cycle functions), using algorithmic examples in the block-based...
and Wiring programming languages code: i) for teaching outputs, we used three LED instances to simulate the functioning of a semaphore; ii) for teaching input, we use buttons for turning on and off a LED that receive values from input. In the Appendix VII, Figures 6(a) and 6(b) illustrate the design and code of the first example (considering block-based and Wiring code); Figures 7(a) and 7(b) depict the design and code of the second example (only with Wiring code). Such as the work of [30], we applied a goal-based evaluation approach for developing programming competencies. In each online class, we gave homework like the implemented exercises in classes.

As the loop function represents an infinite cycle, students started working with cycles from the beginning, and each new example and exercise permitted the development of that algorithmic strategy. Likewise, from the beginning, the students became familiar with the use of subroutines; we explained and highlighted their main purpose and the relevance of dividing a problem into subproblems as well as solutions into sub-solutions for solving sub-problems.

The final homework of the course concerns the use of sensors and actuators, for which we also define ranges of values and conditions for reacting regarding the current sensed value. We asked for modular solutions; students must use functions with parameters that represent common tasks in the solution. We taught about sensors and how the system reacts by turning on and off LED instances regarding the sensed value. Figures 8(a) and 8(b) in the Appendix VII show the circuit design examples in Tinkercad for a Passive Infrared (PIR) and temperature sensor, respectively.

To achieve the learning goals of students, we found that they were far more motivated by watching and interacting with colored led lights going on and off than just reading a classical sentence like “hello world”. Most of them insisted on repeating the experience several times, turning the overall process into an entertaining activity. Though there is no direct screen to display the outputs of an Arduino system, both the Tinkercad simulation and the potential experience with physical circuits allow them to achieve a positive and necessary visual effect for developing their programming competencies.

### IV. RESULTS

We compared students’ academic results in both semesters during the last year of the FoP course using Python (2019) and the first year of using Arduino for the same purpose, that is, to develop programming competencies. Table 1 summarizes the statistics results of all students’ final grades for those academic periods. As we mentioned, throughout 2019, a traditional programming language was used, while during 2020, we introduced Arduino and Tinkercad. For its syntax rules, Python represents an evolution respecting other text-based programming languages such as C, C++, and Java. Nonetheless, the results suggest that students’ physical interaction with developed systems motivates them to develop knowledge and associated competencies in practice. The academic institution grades in Perú are on a scale of 1 to 20 points with a 60% requirement for approval.

According to the results in Table 1, there are a number of significant improvements in the obtained academic grade of students of ISI-UC in using Arduino for developing programming competencies: more than 2 points in 2020 and in each academic semester during that year. Hence, we can appreciate an average improvement in 2020. In both years, professors were the same, and the academic evaluation tools were similar programming projects aiming to develop the same competencies:

1) Using variables.
2) Using conditions and cycles.
3) Using subprograms.

Regarding the development of competencies, each academic semester ends with a final project with the students’ presentation to validate the teaching-learning results. According to these results, there is a clear improvement in 2020. To measure students’ satisfaction in 2020, we conducted surveys about the use of Arduino for programming learning. This study uses a Likert scale of 1 (in-disagreement) and 5 (in-agreement) (Table 2). Table 3 lists the set of answered questions that we classify in those for developing programming competencies (A), those that consider Arduino a motivating tool for learning programming (B), and those for developing competencies in the programming of Arduino (C). The last question about the course material’s usefulness and class is ‘~’. This survey will permit answering this article’s research questions.

The following lines describe part of the survey and the results obtained. They motivate us to continue applying this teaching-learning methodology.

- Arduino for learning programming basics.
  - (i) 89% (72% Strongly Agree and 17% Agree) found that Arduino is a fascinating tool to start programming with.
  - (ii) 76% (41% Strongly Agree and 35% Agree) have managed to understand the concept of variable.

|           | 2019          | 2020          |
|-----------|---------------|---------------|
| Average   | 13.40         | 15.42         |
| Median    | 14.00         | 16.00         |
| Mode      | 13.50         | 20.00         |
| 161 students | 128 students | 128 students |
|           | 2019-1        | 2020-1        |
| Average   | 14.40         | 16.41         |
| Median    | 14.05         | 16.10         |
| Mode      | 14.10         | 20.00         |
| 161 students | 128 students | 128 students |
|           | 2019-2        | 2020-2        |
| Average   | 14.60         | 16.25         |
| Median    | 13.90         | 16.10         |
| Mode      | 14.00         | 20.00         |
| 161 students | 128 students | 128 students |

• Arduino for learning programming basics.
FIGURE 3. Survey results about using Arduino for learning programming basics.
FIGURE 4. Survey results about using Arduino as a motivating element to continue studying it.

(iii) 72% (41% Strongly Agree and 31% Agree) have learned to differentiate and make use of logical operators.
(iv) 62% (41% Strongly Agree and 21% Agree) have managed to understand the use of conditional structures.
(v) 72% (41% Strongly Agree and 31% Agree) have managed to understand the use of repetitive structure.
(vi) 68% (38% Strongly Agree and 28% Agree) have managed to understand the use of subroutines.

- Arduino as a motivating element to continue studying it.
- (i) 76% (14% Strongly Agree and 62% Agree) feel ready to carry out a project with Arduino.
- (ii) 69% (28% Strongly Agree and 41% Agree) believe that it is an ideal platform to solve real problems.
- (iii) 86% (55% Strongly Agree and 31% Agree) plan to purchase an Arduino kit for their personal projects.
- (iv) 87% (52% Strongly Agree and 35% Agree) plan to take a more advanced Arduino course.

Figures 3 and 4 depict the mentioned results. In each graph, the green area represents a strong agreement with the statements presented on the survey; the light green area represents an agreement; the white area shows the neutral responses; the orange area indicates a disagreement with the statements, and the red area represents a strong disagreement. In summary, high student satisfaction exists concerning the use of Arduino for learning programming. We can appreciate the benefits of using Arduino to develop programming competencies and primary skills in electronics. Thus, due to the project's success, professors at the ISI-UC will continue using Arduino for their introductory programming courses.
TABLE 3. Academic survey for students in 2020.

| Class | Question |
|-------|----------|
| A     | Arduino seems a fascinating tool to start programming with. |
| C     | I understand the structural concept of a program for Arduino (void setup() and void loop()). |
| A     | I understand the concept of variables, and how to declare and assign a value to them in the Arduino programming language. |
| C     | I understand the existing primitive data types in the Arduino programming language (int, char, long, float). |
| A     | I can differentiate the types of logical operators (NOT (!), AND (&&), OR (||)) and the way to apply them in the Arduino programming language. |
| A     | I understand the conditional structures if, else, and else, their concept and function in the Arduino programming language. |
| A     | I understand the repetition structure for, its concept and function in the Arduino programming language. |
| A     | I understand the repetition structures while and do..while, their concept and functioning in the Arduino programming language. |
| C     | I understand the semantic, syntax, and practical use of functions analogRead(..) and digitalRead(..) for reading analog and digital input values in the Arduino programming language. |
| C     | I understand the semantic, syntax, and practical use of functions analogWrite(..) and digitalWrite(..) for writing analog and digital output values in the Arduino programming language. |
| C     | I understand the semantic, syntax, and practical use of the function Serial.begin(..) to communicate programs in the Arduino programming language and the PC. |
| B     | I feel ready to do a project with Arduino. |
| B     | I believe that Arduino is an ideal platform to solve real problems. |
| B     | I would like to purchase an Arduino kit and do personal projects. |
| B     | I would be interested in attending a more advanced Arduino course. |
| -     | The course material (presentations, exercises, etc.) was suitable. |

The development of programming skills is a topic of comprehensive study and with different work areas and research in world literature. Tsai [31] mentions the complexity of learning programming languages with complex syntactic and semantic structures and highlights the positive effects of visual programming languages for the development of these skills. In the context of teaching-learning methodologies in programming courses, [32] present results of applying an inverted class methodology to develop programming skills.

Taking into account the results obtained, RQ1 and RQ2 are fully satisfied because Figure 3 shows that students consider Arduino and Tinkercad adequate tools for developing programming competencies (over 73% in average). Likewise, RQ3 is also fully satisfied because Figure 4 presents that students are highly motivated to continue studying and using Arduino in the future (79% in average).

V. DISCUSSION

As the studies of Vidal-Silva et al. [17], [18] highlight, several countries implement programming courses to develop algorithmic skills and programming competencies from early ages. One of the inherent challenges of developing programming competencies is to make the teaching-learning process practical and attractive [33]. Applying Arduino and Tinkercad seem adequate to developing electronic and programming competencies by block-based programming language to emulate the functioning of Arduino solutions, and then, to implement the Arduino circuit physically, to generate the code for Arduino, and put it in Arduino circuit to run the solution. Despite the importance of developing electronic computing skills in elementary school students, the works that address that goal are not abundant in the literature. The works of [17] and [18] trigger the discussion about teaching...
technological tools such as Scratch and Arduino in primary schools are relevant and practical for developing electronic and programming competencies in Chilean primary school children.

Arduino presents multiple benefits for applications in educational environments. Such as Jang et al. [34] discuss, when students design their system, the use of Arduino in the teaching-learning process permits improving design and programming competencies. In a study on the use of Arduino for teaching programming in undergraduate students, Perenc et al. [22] showed that Arduino permits generating high engagement in students and improving the attractiveness of the course. Serrano-Pérez and Juárez López [35] showed that the integration of free educational software and low-cost electronic and mechanical devices as teaching and learning tools promotes increasing confidence, performance, and interest in engineering students. Finally, because of the affordable price and availability of Arduino, students without practical skills or knowledge in electronics can create simple applications beyond their capabilities in traditional console-based programming courses [36].

Lineros et al. [25] and Vidal Silva and Pavesi Farriol [18] highlight that developing electronic and programming competencies for everybody can be possible by the use
of Arduino. That process seems highly acceptable in high-education students because they can apply Arduino in everyday life, considering their academic level and experience. For example, students can measure their variables with sensors and turn on and off LED systems at home. In the Universidad Católica del Norte, Chile, we are currently offering Arduino and Tinkercad sessions for entrepreneur students of different majors, non-directly linked to information and computer science. The obtained results so far only prove that Tinkercad and its block-based programming language is a great tool for starting developing projects. Those experience are part of our next article about the impact of Arduino for developing electronic and programming competencies in entrepreneurs students.

VI. THREATS TO VALIDITY
This work presents relevant results regarding using Arduino and Tinkercad to develop high-education students’ electronic and programming competencies effectively. Nonetheless, it is necessary to discuss the following practical issues:

- Internet requirement: students need internet for using Tinkercad. It was a real issue since some students of ISI-UC were from hills and mountains in Perú with low-level internet access. That was a real issue because the classes were online for the pandemic time. That issue would not occur in traditional classes.
- Arduino requirement: students require Arduino and electronic devices for implementing their emulated solutions physically. That was a requirement overall
for developing electronic competencies. Since the course looks for developing programming competencies, we only gave extra points to those students who physically implemented their solutions. Almost 50% of the students correctly presented physical solutions. That requirement would not be optional in traditional classes.

- Physical emulation and abstraction: we can expect an unbalanced abstraction capabilities development for the emulation and application of electronic components. Even though Arduino and Tinkercad require using the main functions of Arduino solutions, that is, the setup and loop functions, all the other programming components of traditional structured and object-oriented text-based programming languages are here. However, we need to apply those abstractions without forgetting the functioning of the Arduino main modules.

**VII. CONCLUSION**

According to the presented work and the results obtained, we conclude that:

- Arduino represents a highly suitable tool for teaching initial programming courses thanks to its stimulating and concrete visual learning properties.
- Tinkercad facilitates algorithmic and programming skills because it permits programming using a block-based programming approach like Scratch and a Wiring code editor. Tinkercad also allows

```c
int pinLed = 9;
int pinBoton = 8;
int OnOff = 0;

void setup(){
  pinMode(pinLed, OUTPUT);
  pinMode(pinBoton, INPUT);
  digitalWrite(pinLed, LOW);
  Serial.begin(9600);
}

void loop(){
  if (digitalRead(pinBoton) == HIGH){
    OnOff = !OnOff;
  }

  if (OnOff)
    digitalWrite(pinLed, HIGH);
  else
    digitalWrite(pinLed, LOW);
  Serial.println((String) "OnOff: " + OnOff);
}
```

**FIGURE 7.** Circuit design and code example in Tinkercad for using a push button to turn on a LED.
appreciating the syntax and semantic equivalence between block-based solution and Wiring code. Thus, students evolve in the development of programming and Arduino solutions.

• This study showed that the current generation of students feels very comfortable using Arduino and Tinkercad to develop programming and Arduino solutions online. That represents a significant achievement in the field of online education.

In summary, from an academic viewpoint, this article satisfactorily answered each proposed research question highlighting the usefulness of Arduino and the Tinkercad web platform in the collaborative professor-students teaching-learning process for developing programming and electronic competencies. Hence, this experiment demonstrated the feasibility of teamwork in online education. As future work, we expect to continue using Arduino and Tinkercad for developing more advanced programming competencies such as data structures and object-oriented programming.

APPENDIX

APPLYING ARDUINO AND TINKERCAD

See Figs. 5–8.
ACKNOWLEDGMENT
The authors are currently living in a difficult time, plagued by a pandemic. Regardless of that, they continue working as a team to teach their students. Their special thanks to the Universidad Continental in Huancayo-Perú, the Universidad Católica del Norte in Antofagasta-Chile, the Universidad de Tarapacá in Iquique-Chile, and the Universidad Estatal de Milagro in Milagro-Ecuador for assisting them during the whole process and providing them with the budget to fulfill their educative tasks. They feel very encouraged to continue dealing with these educational activities in the coming years.

REFERENCES
[1] C. L. Vidal, C. Cabezas, J. H. Parra, and L. P. López, “Experiencias prácticas con el uso del lenguaje de programación scratch para desarrollar el pensamiento algorítmico de estudiantes en chile,” Formación Univ., vol. 8, no. 4, pp. 23–32, 2015.
[2] K. Young, “Children and technology: Guidelines for parents-rules for every age,” IEEE Technol. Soc. Mag., vol. 36, no. 1, pp. 27–29, Mar. 2017.
[3] S. I. Malik and J. Coldwell-Neilson, “Impact of a new teaching and learning approach in an introductory programming course,” J. Educ. Comput. Res., vol. 55, no. 6, pp. 899–919, Oct. 2017.
[4] M. L. V. Ramírez, “Resolución de problemas alegro y objetos de aprendizaje: Una revisión de la literatura,” RIDE. Revista Iberoamericana Investigación Desarrollo Educativo, vol. 10, no. 20, pp. 1–26, 2020.
[5] C. Troussas, A. Krouska, and C. Sgouropoulou, “A novel teaching strategy comprising commercial off-the-shelf components for indoor through-wall communications that are based on anomalous resistivity and generalized Ohm’s law,” Plasma Sour. Sci. Technol., vol. 17, no. 4, pp. 275–278, 2018.
[6] L. H. Gonzalez-Guerra and A. J. Leal-Flores, “Potentializing the problem-solving competence in programming courses through a practice-based learning+ tutoring strategy,” in Proc. Global Eng. Educ. Conf., 2020, pp. 691–697.
[7] U. Continental. (2021). Ingeniería en Información y Control de Gestión - Universidad Continental. Accessed: Aug. 2, 2021. [Online]. Available: https://ucontinental.edu.pe/carrera/ingenieria-sistemas-informatico/
[8] I. Russell, C. P. Rosiene, and A. Gold, A CS Course for Non-Majors Based Arduino Platform, New York, NY, USA: Association for Computing Machinery, 2020, p. 1309. [Online]. Available: https://doi.org/10.1145/3328778.3372595
[9] C.-Y. Tsai, “Improving students’ understanding of basic programming concepts through visual programming language: The role of self-efficacy,” Comput. Hum. Behav., vol. 95, pp. 224–232, Oct. 2019.
[10] I. G. Mikellides and A. Lopez Ortega, “Challenges in the development and verification of first-principles models in Hall-effect thruster simulations that are based on anomalous resistivity and generalized Ohm’s law,” Plasma Sour. Sci. Technol., vol. 28, no. 1, Jan. 2019, Art. no. 014003.
[11] C. Vidal Silva and L. Pavesi Farriol, “Desarrollo de un sistema de adquisición y tratamiento de señales electrocardiográficas,” Revista Facultad Ingeniería-Univ. Tarapacá, vol. 13, no. 1, pp. 39–46, 2005.
[12] C. Vidal-Silva, J. Serrano-Malebrán, and F. Pereira, “Scratch and arduino for effectively developing programming and computing-electronic competencies in primary school children,” in Proc. 38th Int. Conf. Chilean Comput. Sci. Soc. (SCCC), 2019, pp. 1–7.
[13] M. Tupac-Yupanqui and L. Pavesi Farriol, “Electrónica para todos con el uso de arduino: Experiencias positivas en la implementación de soluciones hardware-software,” Revista Facultad Ingeniería-Univ. Tarapacá, vol. 13, no. 1, pp. 39–46, 2005.
Cristian Vidal-Silva received the B.S. degree in information engineering from the Faculty of Engineering, Catholic University of Maule, Talca, Chile, in 2003, the M.S. degree in computer science from the University of Concepcion, Concepcion, Chile, in 2007, the M.S. degree in computer science from the Michigan State University, East Lansing, MI, USA, and the Ph.D. degree in software engineering from the University of Seville, Seville, Spain. From 2004 to 2007, he was a Research Assistant with the Engineering Faculty, University of Concepcion, and a part-time Professor with the Catholic University of Maule. From 2008 to 2013, he was a Professor with the Catholic University of Maule. From 2014 to 2018, he was a Professor and a Researcher in software engineering and computer science with the University of Playa Ancha, Vina del Mar University, and Autónoma University of Chile. He was a Fulbright Scholar with Michigan State University. Since 2019, he has been with the Administration Department, Economy and Administration Faculty, Catholic University of the North, Antofagasta, Chile. He is the author of more than 30 articles regarding feature-oriented and aspect-oriented software engineering, machine learning, and programming for kids. He is a Reviewer of Scopus and WOS journals.

Leopoldo Pavesi-Farriol has a formal background in electrical engineering and graduate studies with a specialization in automatic control. His expertise during the last 30 years includes the development of software for real-time applications related to industrial systems and automated equipment, management in information systems, and teaching and research in the field of applied information technologies for manufacturing applications. His experience spans several aspects of engineering activities, from the feasibility stages through the specifications requirements, design, development, and testing of projects. He successfully led teams in developing and managing project teams. He was responsible for the technology, execution, budget, and schedule to meet company’s expectations, especially in the IT industry. As part of his academic career, he has contributed toward writing several publications related to industry automation, neural networks, information technologies, collaborative environmental modeling for viticulture applications, and computational intelligence.

Aurora Sánchez Ortiz received the M.Sc. degree in industrial engineering from Universidad de Chile, Chile, and the Ph.D. degree in information science with a minor in management information systems from the University of North Texas, USA, in 2003. She is currently an Associate Professor of information systems and the Director of the Information Technology Management Research Center, Business School, Universidad Católica del Norte, Chile. She is the Director of the Collaboration Network in Information Technology and Systems in Chile (RedSTI). She has published in journals such as Sustainability, Journal of Theoretical and Applied Electronic Commerce Research, Formacion Universitaria, and Academy of Management Journal. She also has contributed academic articles in Springer books and conferences such as AMCIS and Conf-IRM. Her research interests include e-government, smart cities, and digital transformation. She received a Fulbright Scholarship to pursue her Ph.D. degree.

Jesenia Cardenas-Cobo received the Diploma degree in higher education by competences from the Technical University of Ambato, Ecuador, the B.S. degree in information systems from the Escuela Politecnica del Litoral (ESPOL), Ecuador, and the M.Sc. degree in business administration from the Business Technological University of Guayaquil, Ecuador. She is currently pursuing the Ph.D. degree in software engineering with the University of Seville, Spain. She is currently a full-time Professor and the Dean of the Faculty of Engineering Sciences, State University of Milagro (now, UNEMI). She has more than 20 years of professional experience in higher education. Her main research interests include software products and artificial intelligence applied to engineering education.

Felipe Pereira is currently a professor of Spanish and a doctorate in linguistics, and a Specialist in understanding and producing multimodal discourses, comics, and new literature. For many years, he worked as a teacher at various national and international universities, especially on specialty courses and didactics and methodology. He has studied the use of games and video games as learning resources. He had designed and directed a diploma in playful strategies for learning for three years and participated as a teacher in a master with a course on educational innovation.