# Capítulo 3

# Programming by Blocks

Irene Hernández Ruiz and Carolina Gómez Fernández

***Abstract***

*The following workshop describes the methodology for using "block programming" in teaching programming, in which the main concepts in programming such as the use of statements such as conditional variables, cycles, among others, are presented. To achieve this purpose, we will work analyzing the problems from the computational thinking using the tool developed by MIT called Scratch. This programming environment has the advantage that it does not consider the problems of syntax, but, through blocks of different colors, workshop participants can associate a color with a function in programming, making the concepts can see them applied in a visual and striking way to learn. For this work, the proposal of the workshop is presented, which is aimed at a population that does not have programming knowledge but wants to start in this world.*

## 3.1 Introduction to programming

To start learning to program it is necessary to make an analysis of the development environment as well as the programming language, to know which would be the most viable for students to interact within this process and learn in the best way. For example, for Revell "A typical text-based programming language makes the programmer think like a computer, a visual programming language allows the programmer to describe it as a collection of graphics intended to solve an event" [Revell, 2018].

Visual programming languages allow people using this methodology to have a more graphical way to solve problems, using programming elements that can be displayed as images or words in a development environment as Boshernitsan and Downes (2004), point out "such languages are characterized by their reliance on visual techniques throughout the programming process. The programmer manipulates icons or

 

other graphical representations to create a program, which is then debugged and executed in the same visual environment".

This workshop introduces the main programming concepts using a graphical language such as Scratch, which is explained in Section 1.3, and to achieve this, the computational thinking explained in Section 1.2 is applied 1.5.

## 3.2 Computational Thinking

Wing (2006) indicates that the "Computational thinking involves problem solving, system design and understanding of human behavior using the fundamental concepts of computer science". To implement its resolution, computer science resorts to the area of programming, which is the necessary set of instructions given to a machine to be able to perform a specific task. Unlike digital literacy, which involves acquiring a series of basic skills in the use of hardware and software, languages and new forms of communication, digital literacy is not a new skill [UNESCO, 2021].

The idea of Computational thinking unplugged refers to the set of activities, and their educational design, that are developed to encourage children in the early stages of cognitive development (early childhood education, first stage of primary education, games at home with parents and friends, etc.) skills that can then be evoked to promote and enhance a good learning of thinking in technical, professional or university training, including. Activities that are usually done with index cards, cardboard, parlor or playground games, mechanical toys, etc. An interesting case is CS Unplugged[12] is a collection of free didactic material that teaches computer science through interesting games and puzzles with the help of cards, strings, crayons, and lots of physical activity, created by the University of Canterbury, New Zealand.

Problem solving is the conclusion of a broader process that has as previous steps the identification of the problem and its modeling, and to achieve it in computing, algorithms are generated that include the sequence of steps to solve an algorithm.

Schoenfeld (2016) indicates that there are five aspects involved in problem solving: background knowledge, solving strategies, management and control, beliefs and affect, and practices. Management and control, understood as self-regulation and monitoring of the solver, belong to a broader category that we call metacognition. Metacognition refers to the ability to reflect on, understand and control our own learning [Ullauri and Ullauri, 2018]

Problem solving has been positioned in a relevant place, due to the importance it has for the development of competencies for life [English and Gainsburg, 2016]. In addition to understanding the problems, Santos-Trigo (2014) considers it necessary to find diverse ways to interpret, symbolize and discover solutions to the questions posed, which allow to discuss the results found.

García (2017) indicate that the concept of methodology is defined as the set of actions that must be followed to achieve certain objectives previously having some

12 . https://csunplugged.org/es/

 

knowledge on the subject and propose the following problem-solving methodology consisting of the following steps:

- Identify the problem: In this phase, the problem is understood and data that help to understand the problem are highlighted.
- Suggest (or propose) solution alternatives - Appeals to the developer's ingenuity in proposing possible solutions to the problem.
- Design the algorithm - The result of this stage is to create a series of steps that will serve to solve the problem.
- Develop the solution and check the results -In this phase the results are checked.

System design is the process of defining the architecture, modules, interfaces, and data of a system to satisfy previously specified requirements. Systems design could be seen as the application of systems theory to the development of a new product. Currently there are many methodologies for the creation of a system, one of them is design thinking, a dynamic technique to generate innovative ideas that focuses its effectiveness on understanding and providing solutions to the real needs of the users. It comes from the way product designers work. This methodology considers analysis and reasoning as important as intuition for problem solving. It allows to build ideas based on function and emotions. This methodology revolves around the user experience. Its application generates solutions that otherwise could not have been implemented, allowing developers to put themselves in the shoes of their own customers or users [Foster, 2021].

## 3.3 Scratch

Scratch[13] is a visual programming language developed by the Lifelong Kindergarten Group of the Massachusetts Institute of Technology (MIT) Media Lab in 2005, which allows to explain in a playful way the main aspects of programming, allowing a person who wants to learn a pleasant and easy to understand experience. In mid-May 2007, it was placed on the market as a visual programming tool for all ages Pujades (2019).

Scratch is one of the most widely used programming languages to introduce children to the world of computer science, as it has a simple structure and is also attractive for young and elderly people. However, Scratch is not the first initiative created to work as a visual language; prior to this, LOGO is a programming language that uses a pedagogical approach that offers the necessary characteristics for the student to understand the tasks to be performed in the learning process. [Valente, 1997] [Papert, 1999] and that the teacher orients in the teaching approach of comprehension [Andes, 2009] with criteria that facilitate flexible performance in any subject that is known. LOGO provides an environment for students to apply various strategies in problem solving, but this interaction must be mediated by a tutor or teacher. It was developed by Seymour Papert at the Massachusetts Institute of Technology in 1968. One way for children to use LOGO is through a software program called MicroWorlds ™ This graphical software allows children to create their own scenarios (worlds) and moving program icons. The Interaction with the computer for Seymour Papert [Papert, 1999] is

13 https://scratch.mit.edu/

 

to communicate with the computer facilitates the way of production of learning, through programming in LOGO, since students knowing some LOGO instructions such as displacement, rotation, drawing lines and drawings, can be grouped in an orderly manner in procedures to solve a proposed problem.

There are different success stories in the world in which the use of graphic programming languages using Scratch has been motivated, for example in 2015 an experience was conducted on the use of the tool in Chile. The authors found that the use of Scratch is an instrument conducive to the development of logical and algorithmic thinking in students, it also presents an environment that is motivating and allows participation in the proposal of solutions to the situations posed, which enables the analysis of problems, the proposal, development, and implementation of logical and algorithmic solutions [Vidal, 2010].

Another case study is presented by Armoni, Meerbaum-Salant and Ben-Ari, who analyzed the transition from studying computer science with the Scratch visual environment to studying computer science with a professional text programming language such as C# or Java in high school. "We found that the programming knowledge and experience of students who had learned Scratch greatly facilitated learning the more advanced material in high school: less time was needed to learn new topics, there were fewer learning difficulties, and they achieved higher cognitive levels of understanding of most concept" [Armoni, Meerbaum-Salant, Ben-Ari, 2015].

Bustillo Bayón (2015) observed that the inclusion of Scratch in primary education facilitates the incorporation of new learning techniques, teaching methodologies and resources. The students of the Magisterio de Vitoria-Gasteiz, located in Spain, carried out different experiences related to the programming language, which enabled the possibility of including the tool as one of the practices applied by professors when teaching, and allowed the Scratch tool to help teachers to learn, experiment and observe the results that this type of activities generated in their students.

The Colombian Ministry of Education [Bolaños, Cuero and Villalobos, 2017] incorporated the Scratch tool into the mathematics program, to enrich the development of logical mathematical thinking in a fun and problem-solving way. The project was developed with children in third grade of primary school in public schools, where interactive activities on geometric figures were conducted.

In a study conducted by Vázquez and Ferrer of the Spanish Distance University [Gomez, 2010] educational experiences were created with high school students, where they made video games in the classroom using Scratch, obtaining as a result the experience of the students with the ability to design and create complex video games with different programming modules, therefore increasing their technical skills and promoting greater creativity in the teaching-learning process, also obtained as a benefit, that teachers could guide their classes from a more creative perspective.

 

On the other side, Scratch can also be used as an online tool in which you can publish your own work and have access to the work of other colleagues who participate in a community[14].

Also, it is important to highlight that there is currently a version suitable for working with children known as Scratch Jr, which has fewer elements for the development of stories and less text, which is very useful for working with a population of 6 to 8 years old. The following Figure 3.1 shows the Scratch JR environment, which compared to Figure 3.2 shows that it has fewer resources for program design.

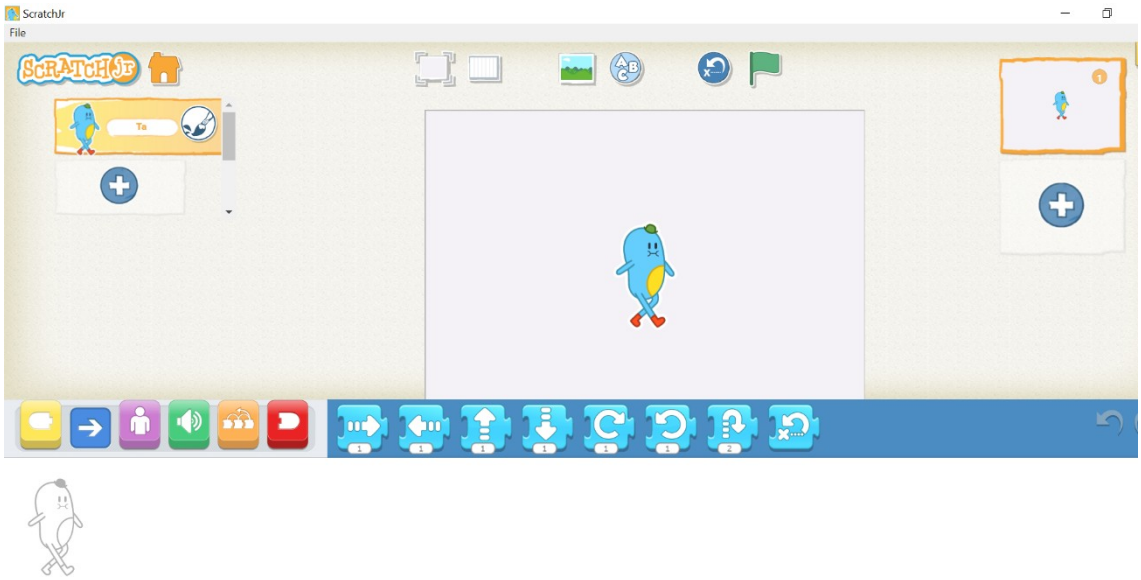


**Figure 3.1. Scratch Jr.**


**Figure 3.2. Top Menu of Scratch**

## 3.4 Human Computer Interaction (HCI)

HCI studies the design, implementation, and evaluation of interactive systems in the context of user activities. In this discipline the term human does not necessarily refer to an individual but can refer to a group of individuals with a given profile or working collectively, in sequence or in parallel. In addition, the term computer refers to a wide range of systems that can range from a desktop computer, a cell phone, a vehicle, a microwave oven, a toaster, an embedded system, to systems that include elements that are not necessarily computerized, such as people or processes. Finally, the term interaction involves everything related to the dialogue between the human and the computer, using input and output devices, either implicitly or explicitly. For these

14 https://scratch.mit.edu/community_guidelines

reasons, when we talk about computers in IHC we are really talking about interactive systems [Castro and Rodríguez, 2018].

Computer design is a process inherent to HCI involving several factors such as: i) the people for whom it is designed (example, their abilities, capabilities, and limitations); ii) the activity to be performed with the computer (example, open heart surgery); and iii) the context in which the activity will be performed (example, an operating room, sitting in the room or in an office or a racing car). Generally, the interaction takes place in a place where the social aspects and organizational context have an important effect on both the person (human) and the system (computer).

Human-Centered Design (HCD) aims to design interactive products that are easy to use, effective in their use and with an enjoyable user experience, as well as to optimize a user's interactions with a system and its environment or product. The HCD focuses on understanding the problem space to propose innovative technology, in general, the design process involves four activities: 1. Identify the needs and establish the requirements for the user experience. 2. Develop alternative designs that satisfy the requirements. 3. Build interactive versions of the designs to be communicated and evaluated. 4. Evaluate the prototype through the process and the user experience. Finally, the usability and user experience of the prototype is evaluated.

One of the reasons for the rise of design in HCI is the need for understanding human needs [Holzinger, 2002], which is associated with aspects of perception, cognition, intelligence, and interaction with any type of information on any type of device [Hooper y Dix, 2013].

### 3.5 Description of the workshop

The block programming workshop is a workshop developed using computational thinking to solve a computational problem using the Scratch tool. This workshop is an initiative to motivate participants to enter the world of programming without the need for prior knowledge.

Among the objectives to be developed for the course are: i. to introduce the concept of programming to the workshop participants, ii. to carry out a practical exercise on the topic of programming, iii. to ensure that the workshop participants can put into practice on their computer the practical exercise developed by the workshop facilitators. To enable the workshop participants to put into practice on their computers the practical exercise developed by the workshop facilitators. This workshop has a duration of 6 hours and is intended for an audience with no previous programming knowledge.

In order to teach programming to children, adolescents and adults, we work as follows: first we describe a problem to be solved trying to relate it to concepts already known by the workshop participants, for example the challenge of developing a calculator with the basic arithmetic operations (addition, subtraction, multiplication, and division), in this way the workshop participants already have in their minds a previous concept, and questions are asked to the participants creating a great participation and

 

that they can get involved in the development of the solution. By having active learning, where the ideal is that the participants have a leading role in the teaching-learning process, other generating questions are created, such as: What are the necessary elements to perform an addition, and then the block programming commands for data input and output are presented. Participants are also asked to present what they are building and choose the numbers needed to perform the tests. Next, the concept of repetition is explained to them, in which the concept of cycle is presented and then they are asked what elements are necessary for the program to work if the user wishes. Keeping the attention of the participants in the educational proposal is what allows the success of the workshop and that those involved can have clarity on when and how to make use of the basic mathematical operators.

## 3.6 Methodology

The methodology used in this project is that of active learning, which is understood as the realization of different activities by the students accompanied by reflection on the actions, they are conducting Bonwell and Eison (1991).

Gómez (2010) indicates that active learning is a method that allows educating, taking the student beyond the passive receiving role and allows the student to have a direction and initiative in his or her education. The teacher's role is to guide students to understand the purpose of the curriculum, to understand what they are doing and what they are doing it for. For Piaget the four principles of learning are: Students must construct their own learning to make sense of the knowledge. They learn best when they can be active and interact and learning is centered on them.

Active learning only requires willingness, creativity, innovation and if possible, experts who can accompany the process [Jerez, 2015], collaborative work among colleagues enriches the whole learning process, not only learning in class but also during the interaction with other colleagues and during the research process to solve problems. All this as part of the active learning process, which is framed within the constructivist theory.

The use of the active learning methodology is proposed, as a mechanism where knowledge is centered on the students, where awareness of the importance of generating a study methodology is raised. This methodology allows the teacher to perform more practical activities where the student is taught the analysis of problems using basic programming structures.

According to Wesley and Richard (2006), indicate that some of the guidelines that can serve as a guide for the teacher in the classroom for the application of active learning are:

- Ask questions to the class during lectures to stimulate curiosity.
- Use guiding questions.
- Use visual tools.
- Teach principles of critical thinking as you teach the subject matter.
- Encourage your students to get to know each other.

 

- Call on all participants.
- Encourage independent thinking.

For this workshop, participants will perform a practical exercise using the Scratch platform. During the workshop, participants will be able to ask questions about the topics covered during the workshop and a set of activities will be developed, which will be explained in detail in Section 1.6 Workshop Structure of this document.

During the development of the workshop, it is necessary to have the following materials: computer with internet connection and access to the Scratch version 3 download for the development of the dynamics. Able to use it from the web page, as well as from the desktop application; once you have logged in, you can start or create an account, which will allow you to store your projects Gough (2018).

## 3.7 Workshop Structure

The following is an explanation of each of the sections that will be worked on during the workshop. In each of the chapters there are the necessary didactic resources to achieve each of the objectives.

### 3.7.1 Installing the program

To work in Scratch, you can download the software from the MIT web site https://scratch.mit.edu/ or you can create an online account.

Below is an image of how the top menu of the tool is displayed in which if a person wishes to create an online account, he/she must click on the "log in" option".

After logging in or having clicked on create, the top bar will show some actions that can be performed, such as: change the language or name, save, or add a previously made project, view tutorials and exit the account if you are logged in.

Once the tool is installed or the session is active, the Scratch environment will be displayed (Figure 3.3).

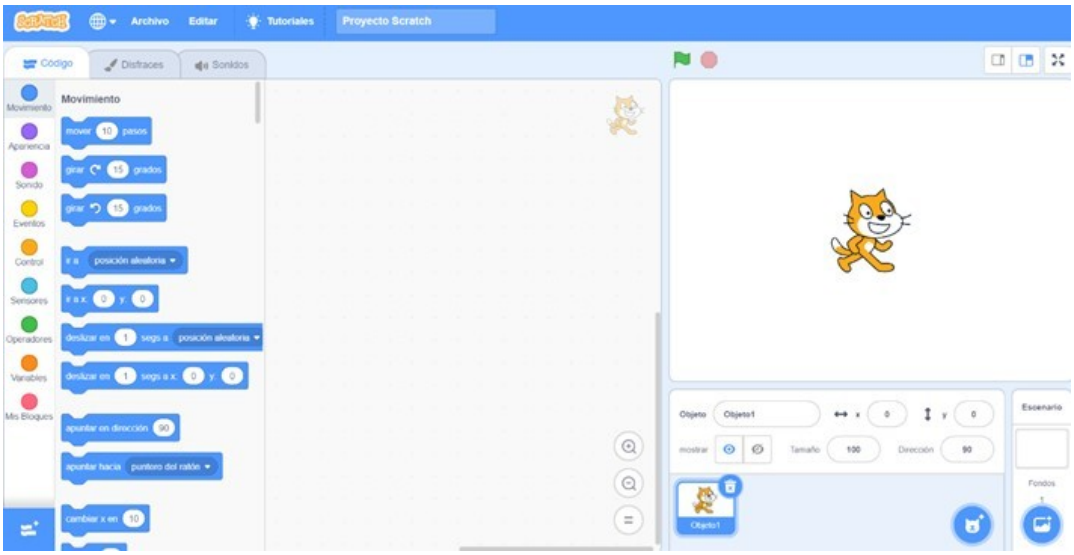


**Figure 3.3. Image of Scratch**

Figure 3.4 shows the Scratch sections. Section 3.7.2 will explain each section in detail.

There are different versions of Scratch, it is important to check that the version installed on the computer is version 3.6.0 so that all the projects can work correctly.

### 3.7.2 Explanation of the environment

The development area is divided in three from left to right; the first one is for the selection of code blocks (orange color), costumes and sound; the second one shows the programming developed in each object (blue color) and, finally, the third one shows the visual part of the program (yellow color).

In the code block, it will allow you to perform some actions, which are divided into categories Castillo (2019):

- Movement: Allows you to rotate and move objects.
- Appearance: Modify the appearance of the object and the background.
- Sound: Sounds can be added, removed, and modified.
- Events: Execute actions determined by the user.
- Control: In which conditionals are included: if, else, forever, repeat and more.
- Sensors or detectors: Allow objects to interact with the user through the different peripherals.
- Operators: Generate numbers randomly, include mathematical operators, among others.
- Variables: Add and modify variables.
- My blocks: In this part, you will find the code blocks that are made by each person and that you want to store.

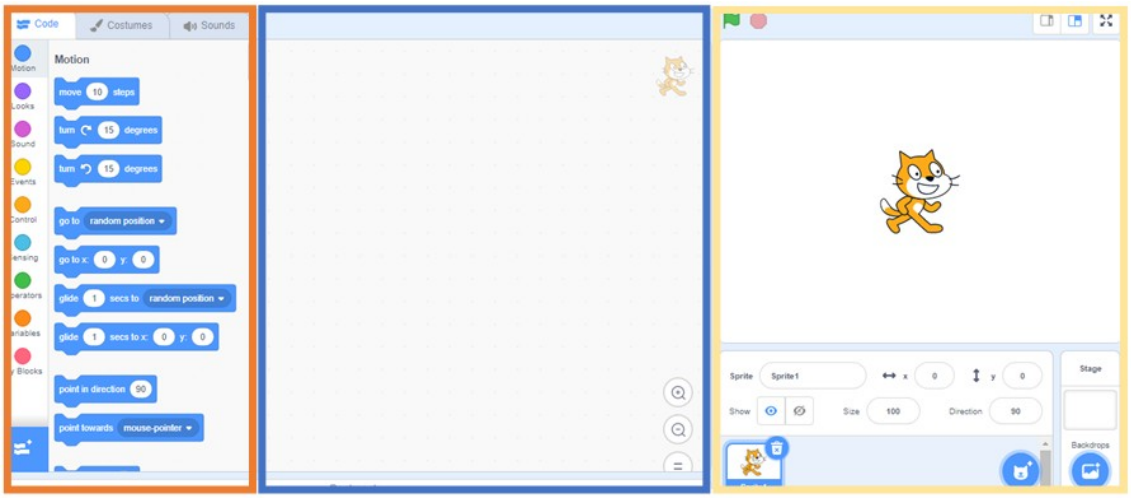


**Figure 3.4. Scratch Environment Description**

Figure 3.5 shows an example of the section in which it is observed that, when the green flag is included, the object is asked to move 10 steps, then to say "Hola!" and wait 2 seconds and then wait 5 seconds for the effect and to rotate 180 degrees to

achieve the effect that the object is upside down. The option to add a desired background has also been added.

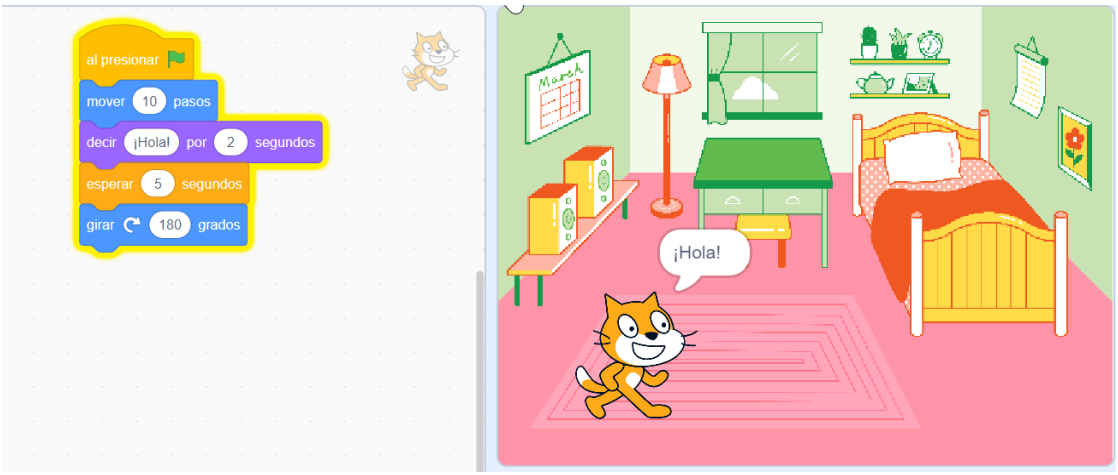


**Figure 3.5. Scratch Exercise**

On the left side you can see the code to perform the exercise, it is recommended to use the following instructions to introduce the subject.

### 3.7.3 Statement of the Problem

Wing (2006), points out that for the development of computational thinking they have the following pillars: i. decomposition of a problem into smaller phases, ii. recognition of repetitive patterns, iii. abstraction of information irrelevant to the proposed problem, iv. written algorithms presented for the resolution of the problem.

In this way, the problem posed is: How can I build a basic calculator, from there the minds of the participants analyze the resources they occupy as explained in the elements that make up the Scratch tool. And it is about decomposing the problem in mini-problems in this way the mini-problems are: i. How do I enter data from the keyboard, ii. What information should I enter, iii. How do I start the program, iv. How do I enter the information from the blocks, v. How do I print an output on the screen, vii. What are the basic arithmetic operations, viii. What elements are needed to perform an addition, ix. If I succeed in performing an addition, what do I need to do to be able to perform the other basic operations, x. What happens with the division by zero (o), xi. How can I repeat the processes?

There are works that indicate how to decompose a problem, for example Computational Thinking leadership toolkit first edition points out these ideas to develop in class, create directions to a location in the school by breaking the directions down into smaller geographical zones. Join the sections of directions together into a whole. Develop a plan to make the school “green.” Separate strategies such as recycling paper and cans, reducing use of electricity, and composting food waste. In planning the publication of a monthly newsletter, identify roles, responsibilities, timeline, and resources needed to complete the project.

 

In order to present the general problem and develop it with the participants, this type of workshop will allow the participants to remember previously learned knowledge as in the case of mathematical operations and to be able to apply concepts such as variables, structures, conditions, cyclic structures and the order of the inputs in the computational problems so that it is a process facilitated by the teacher but that there is a participation of the students within the teaching and learning process.

### 3.7.4 Use of the different resources of the environment for the solution of a problem

First, it is necessary to select the work environment (Figure 3.6) known as scenario and the character that will interact with the development of the problem.

**Figure 3.6. Start**

To start the program, start with Figure 3.7, which indicates that the program will start with the set of instructions placed below this instruction.


**Figure 3.7. Start**

Subsequently, in Figure 3.8, the topic of data entry is introduced with respect to the light blue blocks to send the message and a block in the same color for the response. In this way, the data input is stored in the response, and we proceed with the explanation of the concept of conditional structure. For the corresponding analysis, we proceed to evaluate the input with the mathematical options in this case we first choose the implementation of the sum.

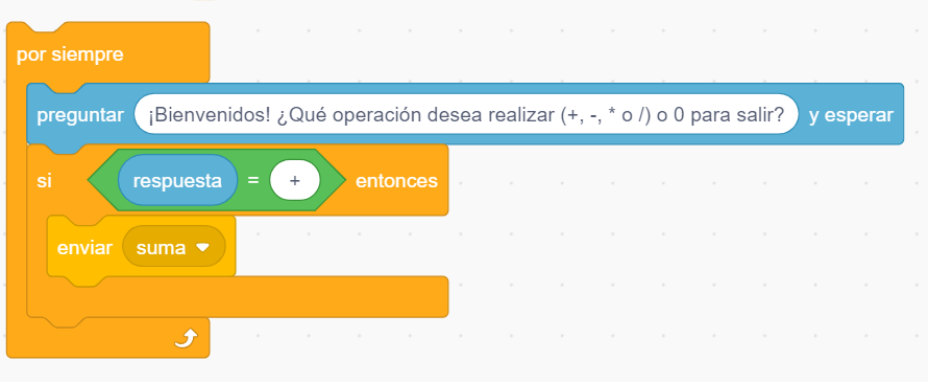


**Figure 3.8. Operation**

The addition option is chosen, as it is a known operation for the participants, in which a subroutine is sent to explain what elements are required to perform the addition. In this way, the first pillar of computational thinking, that of decomposing the problem into mini problems, is applied.

Subsequently, Figure 3.9 explains the case of entering an invalid value to the mathematical operation. It can be seen in the figure that the invalid option is displayed when information is entered that does not correspond to.


**Figure 3.9. Invalid option.**

In the process of the subroutine, it is explained that it is necessary to analyze the problem of addition, and with this new problem posed again, the necessary elements to perform the calculation of a sum are abstracted with the participants. In this way pillar number 3 is applied.

Once the addition has been done, the participants are asked how the other mathematical operations are constructed. It is then that the participants analyze the previous problem and realize that what they must change is the operator and the process must be repeated, thus making it possible to apply pillar number 2.

Figure 3.10, shows the image of adding two numbers and displaying it on the screen, thus making it possible to visualize the corresponding sum.

 


**Figure 3.10. Sum of two numbers**

Figure 3.11 shows that it is possible to add another type of mathematical operation to be performed, as well as to develop the creation of two variables.


**Figure 3.11. Sum of two numbers**

Pillar 4 is applied with them to achieve the implementation of the algorithm and put it to work by making each one of the selected blocks.

Finally in Figure 3.12 you can see how the school scenario and the cat object which is the one that shows the information of the options to be performed and in the upper left part you can see how the numbers change.

 


**Figure 3.12. Sum of two numbers**

### 3.7.5 Presentation of the student's solutions about what was developed in the workshop

A space is created during the workshop so that students can present what they have done during the workshop and socialize some ideas. It is important to point out that the interaction with the participants is fundamental so that each of the questions can be answered in time.

### 3.7.6 Other resources

Other resources are available to the participants, such as Challenge 1.

"Disguises Exercises": this challenge aims to familiarize the student with the first categories studied: events, appearance, sounds and events. Goal to apply in 4 scenarios trivia questions, for example:

Question #1:

Did you know that at the time of the dinosaurs there were no continents, there was only one large land mass called Pangaea?

In this case, the background must first be defined and the desired object 1 must be configured (Figure 3.13).

To see the complete exercise, it is recommended to review the exercise Exercises Costumes and then incorporate the necessary resources such as movement and others.

Figure 3.14 shows the exercise proposed for the analysis of the programming process.

 


**Figure 3.13. Creating a world with the dinosaur**

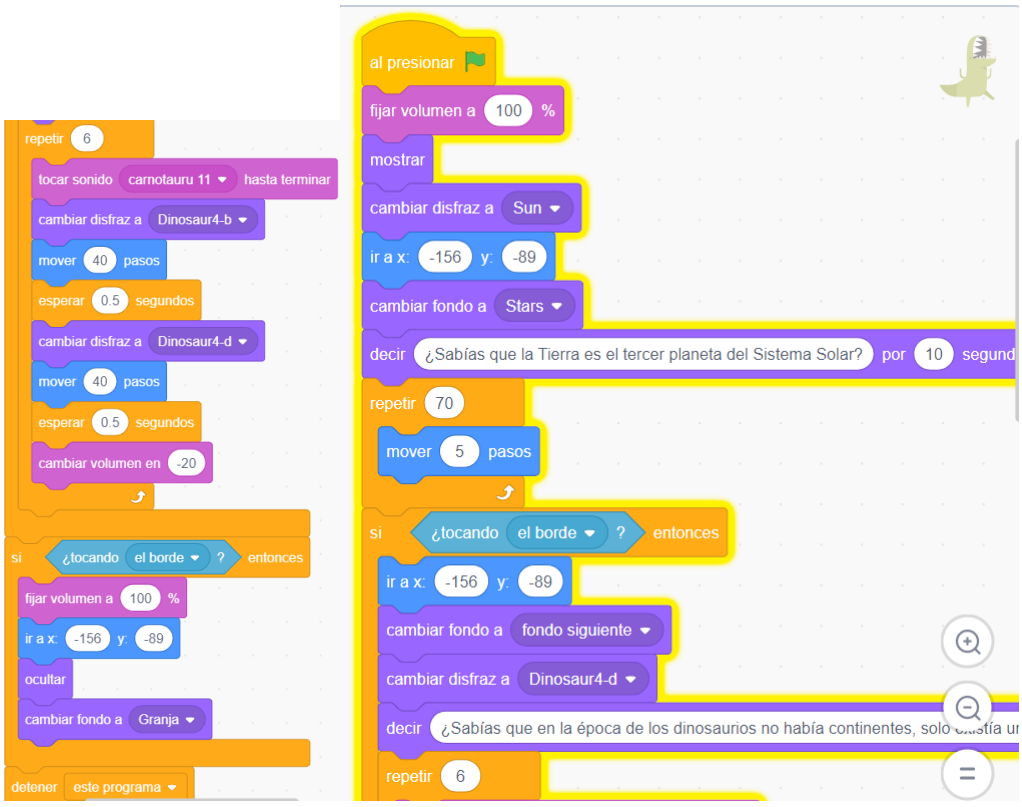


**Figure 3.14. Creating a world with the dinosaur**

Figure 3.15 shows how to work with sound and actions with the edges of the backgrounds. This exercise is intended to further develop logic in problem solving and to motivate the participants.

The sound as one of the resources that Scratch presents for programming is very attractive for the projects. To make use of it, access the sound section where you can

add pre-existing sounds as shown in Figure 3.15 in this figure you can make the sound play faster or slower, as well as sharper or louder. This type of functionality that Scratch presents is very useful because you can give characterization to the story or program to be developed. In addition, this sound can be one's own, in the Figure 3.16 shows the option to record one's own voice by clicking on the microphone option.

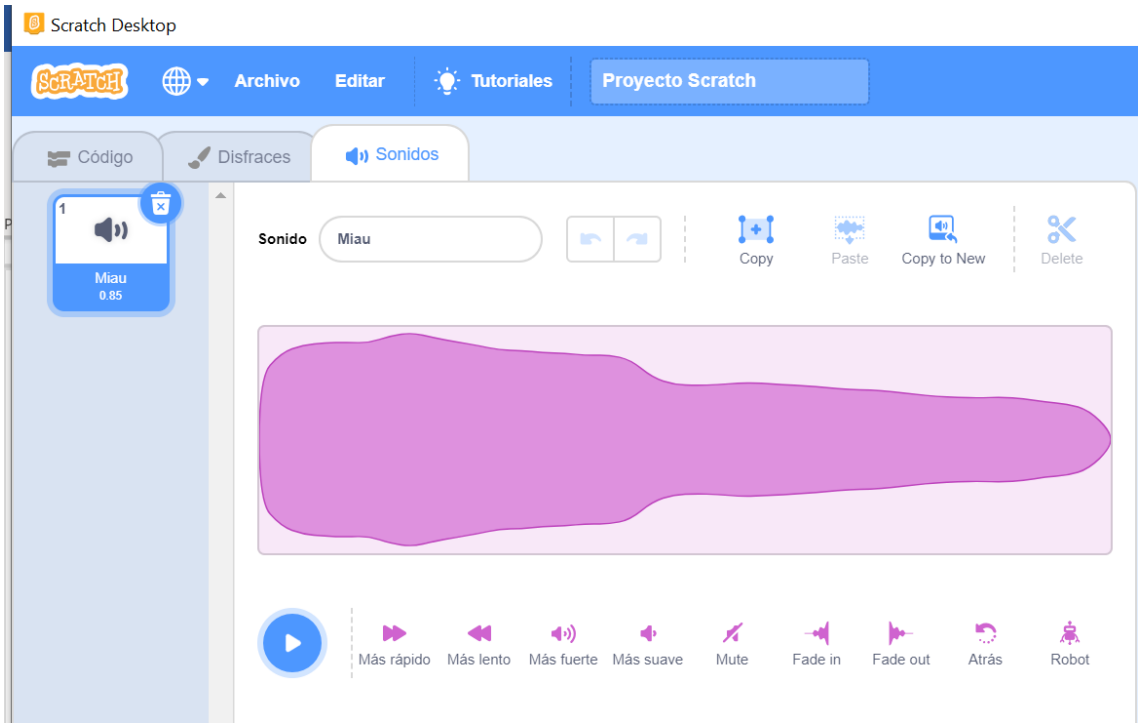


**Figure 3.15. Creating a world with the dinosaur**

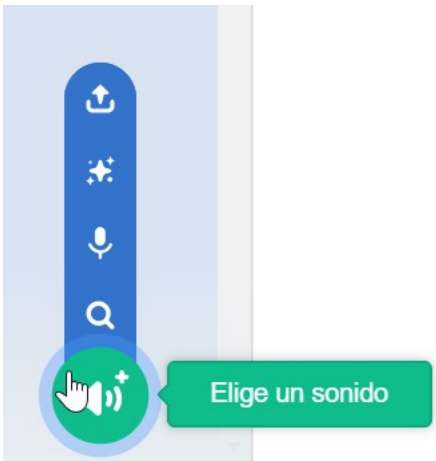


**Figure 3.16. Select the sound**

Subsequently, one's own sound can be included in the story and can be modified as it was done with sound like in Figure 3.17.

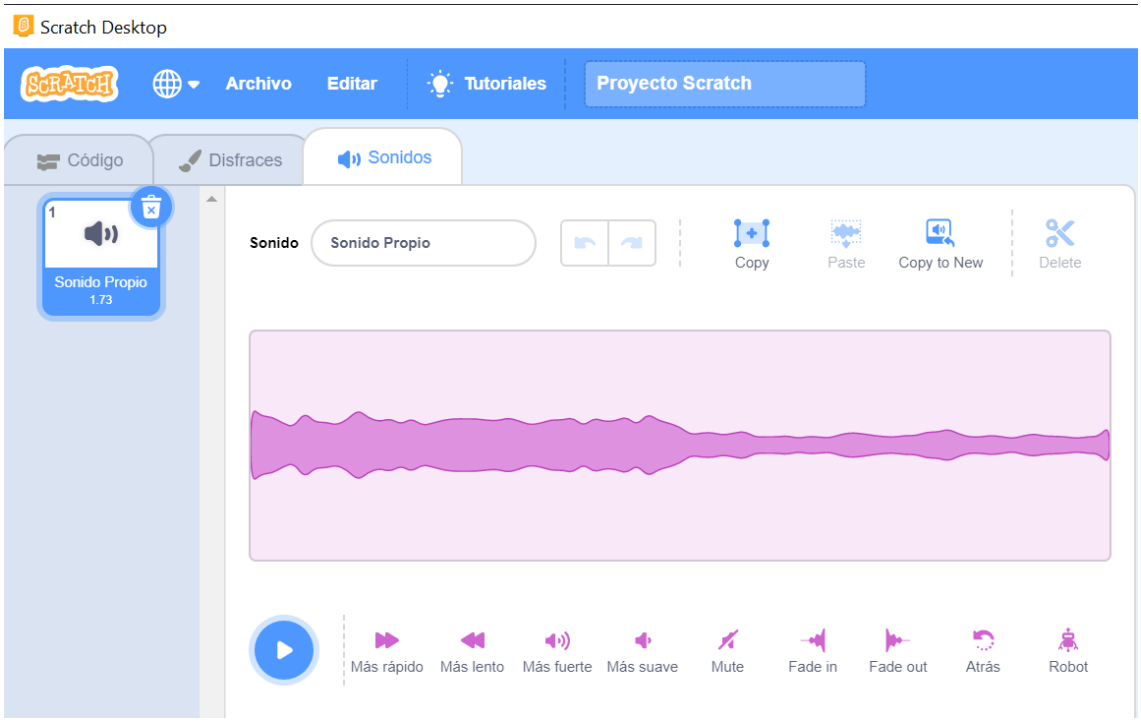


**Figure 3.17. Select the sound**

## 3.8 How to learn more about Scratch?

To continue working with Scratch, it is recommended to make use of different Story Cards, in which you can read and analyze the different stories that have been presented, such as changing scenarios, character changes, character communication, among others. There is also the Scratch community where it is possible to visualize the projects made by other participants, see their creativity, and publish their own ideas. For 2021, at the Scratch Annual Conference, different activities were presented for image recognition in which Scratch objects can be programmed and put to work according to the movements of people, making it possible to generate more interactive projects for the users of this tool.

On the other hand, you can also include some topics in basic electronics to get to know and link the concept of the Internet of Things (IoT), to achieve this it is recommended to review the Arduino websites, as well as review sites like Tinkercard and Microbit where you can program using block programming such as the one presented in this programming workshop with Scratch.

## 3.9 Conclusions and recommendations

The use of block programming allows participants to relate a color to a specific activity or function, allowing them to incorporate instructions quickly.

This type of programming is very motivating for the participants because the blocks are eye-catching, and they can give a characterization of the character and the environment to develop a program.

 

Block programming eliminates the problem of syntax errors in programming, making it possible to concentrate during the workshops on problems to be solved by applying the theme of computational thinking.

Computational thinking can be used to develop solutions to problems in other areas such as STEM (Science, Technology, Engineering, Mathematics) careers to generate in the participants the ability to develop abstraction and logic in problem solving.

Long-term programming should be encouraged to be studied as an input in the current curriculum which can motivate more generations to relate to careers in technology or can help many professionals in other areas that increasingly require more computing within their disciplinary areas.

Among the recommendations provided is that participants can make use of other resources to continue practicing in this area. For this purpose, resources are presented in Section 3.7.6.

Finally, this workshop was intended to be a motivational space in the study of introductory instructions in programming to be a first step in the study of the discipline in an attractive, simple, and functional way.

## References

Armoni, M., Meerbaum-Salant, O. and Ben-Ari, M. (2015). “From zero to actual programming".

ACM. Transactions on Computer Education. (2015). https://doi.org/10.1145/2677087

Andes. (2009). Teaching for understanding. Accessed on 4/5/2009. At: http://learnweb.harvard.edu/andes/tfu/info3d.cfm

Bonwell, Ch. And Eison, J. (1991). “Active learning: Creating excitement in the classroom”. ASHE-ERIC Higher Education Reports.

Boshernitsan, M., Downes M., Boshernitsan and Downes (2004). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.2491&rep=rep1&type=pdf

Bustillo Bayón, J. (2015). “Teacher training with Scratch: analysis of the low incidence in the classroom”. https://www.redalyc.org/pdf/310/31043005010.pdf

Bolaños, M., Cuero E. and Villalobos N. (2017). “Use of Scratch as a tool for the development of mathematical competence.” https://www.researchgate.net/publication/326000060_Uso_de_Scratch_como_herramienta_para_el_desarrollo_de_la_competencia_matematica

Castillo, G. (2019). “What is Scratch and how it works”, https://techlandia.com/13718917/que-es-scratch-y-como-funciona

Castro, L. A. and Rodríguez, M. D. (2018). Interacción Humano-Computadora y Aplicaciones en México. http://www.amexcomp.mx/files/InteraccionHumanoComputadora.pdf

English, L. D. and Gainsburg, J. (2016). “Problem solving in a 21st century mathematics curriculum”. L D & Kirshner, D (Eds.) Handbook of international research inmathematics education [3rd edition] (100 Cases series). Routledge, United States of America, pp. 313-335.

Foster M. K. (2021). “Design Thinking: A Creative Approach to Problem Solving. Management Teaching Review; 6(2): 123-140. doi: https://doi.org/10.1177/2379298119871468

García-Ávila, S. (2017). “Digital Literacy," Reason and Word, vol. 21, no. 3_98, pp. 66-81, Aug. 2017.

Gómez Gómez, J. E. (2010). “Methodological approach for the design of active learning activities supported by contextual awareness”. Doctoral dissertation.

Gough, J. (2018). “Coding with Scratch: An interactive addition quizzer”. Australian Primary Mathematics Classroom, 23(2), pp. 19–22.

Holzinger A. (2002). “Multimedia BasicsLearning”. Cognitive Basics of Multimedia Information Systems, 2. Laxmi-Publications, New Delhi

Hooper C. and Dix A. (2013). “Web scienceand human-computerinteraction: forming a mutuallysupportive relationship. Interactions”, 20(3), 52–57

Jerez Yáñez, O. (2015). “Active learning, diversity and inclusion. Approach, methodologies and recommendations for implementation”.

Papert, S. (1999). Papert on Piaget. Time magazine, pág. 105.

Pujades, N. (2019). “Congratulations! Scratch celebrates 10 years since its creation”, https://www.scratch.school/aprender/cumple-10-scratch/

Revell, M. (2018). “What Is Visual Programming?” https://www.outsystems.com/blog/what-is-visual-programming.html

Santos-Trigo, M. (2014). Problem solving in mathematics education. En S. Lerman (Ed.), Encyclopedia of Mathematics Education (pp. 496-501). Nueva York, NY: Springer. https://doi.org/10.1007/978-94-007-4978-8_129

Schoenfeld, A. (2016). “Learning to Think Mathematically: Problem Solving, Metacognition, and Sense Making in Mathematics”. *The Journal of Education, 196*(2), 1-38. https://www.jstor.org/stable/26612611

Ullauri, J. I. U. and Ullauri, C. I. U. (2018). “Metacognición: razonamiento hipotético y resolución de problemas”. Revista Scientific, vol.3, pp. 121-137. https://doi.org/10.29394/Scientific.issn.2542-2987.2018.3.8.6.121-137

Unesco (2021). https://iite.unesco.org/files/policy_briefs/pdf/en/digital_literacy.pdf

Valente, J. A. (1997). The role of computers in education: skills and comprehension. Perspectives, XXVII (3), 433-446.

Vázquez, E. and Ferrer, D. (2015). “The creation of video games with Scratch in Secondary Education”. Communication Papers, 4, 6, 63-73. (2015)

Vidal, C., Cabezas, L., Parra, J. and López L., (2015) “Practical Experiences for Using the Programming Language Scratch to Develop Algorithmic Thinking of Students in

 

Chile. 2015, vol.8, n.4, pp.23-32. ISSN 0718-5006. http://dx.doi.org/10.4067/S0718-50062015000400004.

Wesley, H. and Richard, P (2006). The Miniature Guide to Practical Ways for Promoting Active and Cooperative Learning (Thinker's Guide Library). The Foundation for Critical Thinking; Third edition.

Wing, J. M. (2006). "Computational thinking". Communications of the ACM, 49(3), Marzo 2006. http://dx.doi.org/10.1145/1118178.1118215

 