Online course for the development of Computational Thinking through ScratchJr using the ADDIE Model

Curso online para o desenvolvimento do Pensamento Computacional através do ScratchJr utilizando o modelo ADDIE

Maribel dos Santos Miranda-Pinto *[1], **Pilar Ibáñez-Cubillas** †[2] and **Slava López-Rodríguez** ‡[3]

[1] *Universidade Aberta, Laboratório de Educação a Distância e eLearning (LE@D), Portugal.*
[2] *University of Málaga, Spain.*
[3] *University of Granada, Spain.*

Abstract

Computational Thinking (CT) is a pivotal 21st-century competency, increasingly integrated into educational curricula through programming and robotics. However, many teachers lack the interdisciplinary training necessary to incorporate CT across various subjects, highlighting a critical need for professional development programs. This article details the design and evaluation of an online course for educators focused on CT and programming via ScratchJr, utilizing the ADDIE instructional model. Structured into five two-hour autonomous modules, the course offers a flexible learning framework. A qualitative case study was conducted during the implementation and evaluation phases, using questionnaires with undergraduate Pedagogy students. Results indicated a significant enhancement in participants' programming and CT skills, alongside a more favorable outlook toward applying these tools in their future practice. Notably, 96.4% and 100% of students positively assessed the integration of these competencies, demonstrating the course's potential to bridge the digital divide in initial teacher training. This initiative was selected for the Scratch Education Collaborative (2022-2024 cohort) and integrated into a MOOC for a Portuguese R&D project, underscoring its relevance in digital literacy.

Keywords: Computational thinking. Programming. E-Learning. Instructional design. Virtual Learning Environments.

Resumo

O Pensamento Computacional (PC) é uma competência fundamental do século XXI, cada vez mais integrada ao currículo educativo por meio de atividades de programação e robótica. No entanto, muitos professores carecem de formação interdisciplinar em PC que lhes permita incorporá-la em várias disciplinas, o que ressalta a necessidade de programas de educação continuada. Este artigo apresenta a concepção e avaliação de um curso online para o desenvolvimento profissional de educadores em PC e programação usando o ScratchJr, empregando o modelo instrucional ADDIE. O curso está estruturado em cinco módulos de duas horas cada, permitindo uma aprendizagem autónoma e flexível. Foi adotada uma abordagem qualitativa por meio de um estudo de caso durante a fase de implementação e avaliação, aplicando questionários a alunos do curso de Pedagogia. Os resultados da implementação inicial revelaram um aumento significativo nos conhecimentos e habilidades dos participantes em programação e informática, bem como uma percepção mais favorável em relação à sua aplicação na futura prática profissional. 96,4% e 100% dos alunos avaliaram positivamente a integração destas competências, evidenciando o potencial do curso para reduzir a exclusão digital na formação inicial. Este curso foi selecionado pela Scratch Education Collaborative para a coorte 2022-2024 e ligado a um MOOC de um projeto português de I&D, consolidando a sua relevância na formação inicial em competências digitais.

Palavras-chave: Pensamento computacional. Programação. E-Learning. Design instrucional. Ambientes Virtuais de Aprendizagem.

*Email: maribel.miranda@uab.pt
†Email: pcubillas@uma.es
‡Email: slavalr@ugr.es

Texto livre
Linguagem e Tecnologia

DOI: 10.1590/1983-3652.2026.64407

Session: Articles

Corresponding author: Maribel dos Santos Miranda-Pinto

Section Editor: Daniervelin Pereira
Layout editor: Leonardo Araujo

Received on: February 13, 2026
Accepted on: April 23, 2026
Published on: September 25, 2026

1 Introduction

In recent decades, Computational Thinking (CT) has emerged as a fundamental skill for 21st-century citizens, evolving alongside advances in programming and robotics and their progressive integration into educational curricula. Since Papert's (1985) seminal proposals, CT has been recognized not only for enhancing higher-order cognitive skills but also for enabling children to explore and reflect upon their own cognitive processes. As Papert (1985) noted, by teaching a computer to "think," children engage in an introspective exploration that transforms them into "epistemologists," fostering a level of metacognition seldom seen even in adults. This framework was later expanded by Wing (2006, 2008), who posited that CT is not confined to computer science but is a transversal skill as essential as literacy or numeracy. Consequently, contemporary literature underscores the necessity of introducing these competencies during the earliest stages of schooling, seamlessly integrated into daily classroom practices.

Among the tools developed for this purpose, Scratch and ScratchJr are prominent as visual programming languages designed to democratize early access to coding. These platforms foster analytical skills, creativity, collaboration, and problem-solving while promoting playful, multisensory, and meaningful learning experiences (Resnick; Rosenbaum, 2013). Specifically, ScratchJr—tailored for early childhood education—empowers young learners to construct interactive narratives and games using block-based sequences, which fosters structured narrative and logical reasoning (Bers; Sullivan, 2018).

Specialized literature has demonstrated that early engagement with programming and robotics promotes the development of vital cognitive and socio-emotional competencies. Such experiences allow children to explore novel ways of representing the world, improving spatial orientation, motor coordination, and observational skills (Bers, 2017). Furthermore, through experimentation and structured play, children learn to collaborate, communicate complex ideas, and reflect on their actions, establishing a robust foundation for early digital literacy.

Authors such as Resnick and Rosenbaum (2013) and Bers, Flannery, *et al.* (2014) emphasize the role of play as a catalyst for learning, as it facilitates experimentation, creativity, and problem-solving within meaningful contexts. In this regard, programming serves not merely as a tool for technical acquisition but as a vehicle for fostering social and emotional learning through a constructivist lens.

Aligned with this approach, Brennan and Resnick (2012) propose a three-dimensional framework for Computational Thinking (CT). This includes computational concepts (e.g., iteration, parallelism), computational practices (e.g., debugging, remixing), and computational perspectives (the evolving self-perception and worldviews of the designers). This holistic vision underscores the urgency of teacher education programs that enable the coherent and contextualized implementation of these dimensions in the classroom.

However, despite the burgeoning scientific literature and institutional recognition of CT, educational initiatives that systematically integrate it into early childhood and primary education remain scarce.Thus, existing research often focuses on isolated interventions, leaving a gap in evidence regarding sustainable training programs that empower teachers to utilize these tools independently.

Addressing this gap, teacher education—both initial and continuing—assumes a strategic imperative. Distance learning environments offer significant opportunities to equip pre-service and in-service teachers with tools like ScratchJr, fostering pedagogical appropriation and practical application. In this context, the ADDIE instructional model (Analysis, Design, Development, Implementation, and Evaluation) emerges as an effective methodology for designing structured, scalable training tailored to educators' needs.

Consequently, this research aims to design, implement, and evaluate an online course based on the ADDIE model, specifically geared toward CT proficiency and programming with ScratchJr. Developed on the Moodle platform, the course has been implemented in postgraduate contexts and selected by the Scratch Education Collaborative (SEC) for the 2022-2024 cohort. Furthermore, its integration into a MOOC within a Portuguese R&D project reinforces its international relevance. Thereby, this study seeks to advance professional development by offering an innovative, research-based pedagogical approach that promotes inclusive, creative, and collaborative learning in early education.

1.1 Literature Review

1.1.1 Programming Language for Early Childhood Education

The introduction of programming languages in early childhood education represents a burgeoning trend in pedagogical innovation, fostering the early development of core cognitive competencies such as Computational Thinking (CT) and problem-solving. These transversal skills facilitate an active and creative engagement with technology from a formative age.

Programming a computer necessitates decomposing complex tasks into sequential steps, formulating precise instructions, and anticipating outcomes—actions that trigger logical reasoning, strategic planning, and metacognition (Espadeiro, 2021). Within this framework, children do not merely acquire new modes of expression; they pivot from being passive consumers of information to becoming proactive producers of digital content. This pedagogical shift underpins the design of visual programming tools like ScratchJr and Scratch, which provide accessible, playful, and meaningful environments that stimulate logical reasoning, collaboration, and self-expression (Roque; Leggett, 2014; Roque, 2022).

From constructivist and sociocultural perspectives, early programming education is anchored in active methodologies that integrate play and experimentation. Resnick and Rosenbaum (2013, p. 165) emphasize the significance of tinkering and bricolage—learning styles defined by exploration, iteration, and discovery. As they posit, "play is a way of relating to the world, testing limits, and experimenting with new possibilities". This reinforces the premise that profound learning occurs when students interact with materials and contexts that empower them to externalize and iterate upon their own ideas.

Empirical evidence underscores the positive impact of programming on both cognitive development and socio-emotional maturation. Such findings have catalyzed the adoption of innovative pedagogical frameworks in formal learning environments. In this vein, Papert (1985) asserted that CT can be introduced even at the preschool level through programming, enabling children to reflect on their own cognitive mechanisms and develop an epistemological awareness of their learning. This metacognitive dimension is vital for fostering autonomy and learner agency.

Similarly, the Positive Technological Development (PTD) model proposed by Bers (2007, p. 6) provides an integrative framework that bridges technology use with holistic child development. According to this approach, embedding programming into early childhood education can cultivate essential psychosocial assets known as the "5 Cs": competence, connection, character, confidence, and caring. These dimensions flourish within a healthy and positive educational ecosystem, where children do not merely acquire technical proficiency but also learn to engage in collaboration, explore personal interests, and engage in creative problem-solving. In a similar vein, Bers (2008) argues that the strategic use of mobile devices, such as smartphones and tablets, require a reimagining of pedagogical approaches at this stage, facilitating the creation of engaging and rewarding learning experiences for young children (Bers; Flannery, *et al.*, 2014).

1.1.2 Programming ScratchJr

ScratchJr is a visual programming language specifically engineered for children aged five to seven, aimed at introducing the fundamentals of Computational Thinking (CT) and digital creation during early childhood (Bers; Resnick, 2016). The platform emerged from an interdisciplinary collaboration between the Developmental Technologies Research Group (DevTech) at Tufts University, the Lifelong Kindergarten group at the MIT Media Lab, and the Playful Invention Company (Bers; Resnick, 2016). Conceived as an adaptation of the original Scratch environment, ScratchJr was tailored to align with the cognitive and psychomotor developmental stages of preschoolers and early elementary students.

Available as a free application for iOS and Android, ScratchJr enables children to construct interactive narratives and animations through a system of graphical blocks representing programmable instructions. Its intuitive interface fosters a hands-on learning environment where pre-literate children can engage with programming, making it uniquely suited for early childhood education.

From a pedagogical standpoint, ScratchJr facilitates more than the acquisition of basic coding concepts, such as algorithms and sequencing; it also cultivates critical thinking and problem-solving prowess. As Bers and Resnick (2016, p. 2) emphasize, "ScratchJr encourages creativity and expression, allowing children to create their own interactive projects through coding". By designing and manipulating block sequences, children learn to decompose problems into sequential steps and build logical frameworks. This process simultaneously reinforces fine motor skills and hand-eye coordination while enhancing socio-emotional competencies, including collaboration and teamwork (Bers, 2008).

Furthermore, programming activities with ScratchJr contribute to both socio-emotional and cognitive development by fostering a playful environment that stimulates curiosity and self-directed learning. Consequently, the use of ScratchJr not only facilitates computational literacy but also empowers children to interpret and reflect upon the world around them in an accessible and engaging manner (Miranda-Pinto; Osório, 2020).

Drawing on Bers's (2022a, p. 140) "Playground" metaphor, ScratchJr provides open coding spaces where, as she states, "Playgrounds invite fantasy play, imagination and creativity, social interaction, and teamwork; they are spaces to experiment with conflict resolution, and they require less adult supervision". Within this framework—which aligns with the ideal conditions for learning to code—Bers (2022a) further posits that:

> In the coding playground, the child is the artist who learns to code, and the paintbrush is the programming language that supports creativity. The colors are the values the child learns and expresses while coding. The coding playground becomes an art studio for practicing a palette of virtues. Creative programming can be a pathway for character development (Bers, 2022a, p. 145-146).

Values such as generosity, honesty, patience, curiosity, optimism, and perseverance, which comprise this "palette of virtues," can be effectively cultivated within ScratchJr coding environments. (See Figure 1).

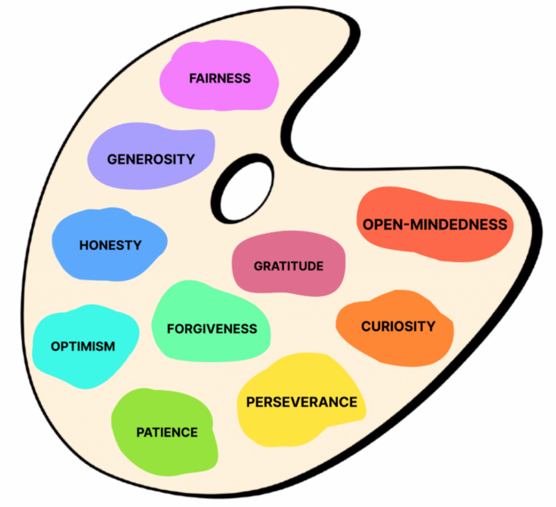

Figure 1. The Palette of Virtues.
Source: Image retrieved from Bers (2022a, p. 147) and also available in: https://sites.bc.edu/devtech/research-2-2/palette-of-virtues-in-practice.

In this regard, the programming environment acts as a catalyst for fostering human values, such as those outlined in Bers's (2022a) Palette of Virtues. It creates a conducive atmosphere for collaboration and teamwork, enabling children to exchange ideas, engage in collective construction, and reflect on their own creative and cognitive processes when approaching problem-solving.

1.1.3 Training in Computational Thinking and programming through e-learning for education professionals

The development of Computational Thinking (CT) has gained significant traction within the scientific community due to its increasing integration into educational contexts. The conceptual evolution of CT has been shaped by a historical trajectory of researchers committed to redefining how we perceive the acquisition of 21st-century competencies. As argued by Wing (2006), these skills should no longer be viewed as the exclusive domain of computer scientists or engineers but rather as essential literacies for all.

Establishing digital literacy is a prerequisite for advancing toward higher-level programming competencies. Learning to code not only facilitates the mastery of a programming language but also fortifies CT and problem-solving abilities across all disciplines. It is imperative that both educators and students develop foundational digital literacy before engaging in computer programming, which necessitates articulating clear, precise commands to execute tasks through structured sequences. In educational settings, the traditional pillars of literacy and numeracy (reading, writing, and arithmetic) should be integrated with CT activities to enhance children's analytical and reflective capabilities.

Integrating CT and programming into early childhood education demands specialized teacher training, encompassing both initial and continuing professional development. In this regard, tools such as ScratchJr are particularly salient, offering a visual, playful, and meaningful introduction to basic coding. However, for effective classroom implementation, teachers must be equipped not only with technical proficiency but also with the pedagogical strategies necessary for its educational application (Bers; Sullivan, 2018; Miranda-Pinto, 2022).

Pre-service and in-service training in Computational Thinking (CT) and programming empowers educators to acquire essential digital competencies that transcend mere instrumental use, focusing instead on creating innovative, inclusive, and student-centered learning experiences. This professional development is increasingly vital as educators must prepare children to thrive in a pervasively digitalized world, where the ability to solve problems in a logical, structured, and creative manner is paramount.

In this context, distance education via e-learning environments offers an effective and accessible alternative for teacher education. As Ibáñez-Cubillas (2018) notes, e-learning in its various forms generates learning opportunities that allow educators to access specialized knowledge without the logistical constraints of traditional face-to-face formats. By offering flexibility in both time and space, this modality aligns with the professional realities of teachers, facilitating their active and sustained engagement in continuous professional development, as evidenced by the positive outcomes reported by Amante, Souza, Quintas-Mendes, Monteiro, *et al.* (2019), Amante, Souza, Quintas-Mendes, and Miranda-Pinto (2023) and Souza, Amante, and Quintas-Mendes (2020).

However, for online training to be effective, it must be rooted in robust pedagogical principles. After all, the quality of an e-learning initiative depends on its instructional design—defined as the systematic planning of objectives, content, activities, resources, and assessment. Consequently, it is necessary to ensure constructive alignment between learning objectives, proposed activities, support materials, and assessment criteria (Hadi *et al.*, 2017). Designing online programs under these principles not only guarantees internal consistency and pedagogical relevance but also amplifies their impact on actual classroom practice, a correlation substantiated by the research of Quintas-Mendes, Souza, and Amante (2022) and Amante, Souza, Quintas-Mendes, and Miranda-Pinto (2023).

1.1.4 ADDIE Instructional Model

Instructional Design (ID), defined as "a systematic method for developing educational and training programs aimed at improving student performance" (Hadi *et al.*, 2017, p. 149), is a cornerstone in the planning, production, and implementation of training initiatives within virtual environments. ID models provide a structured and systematic framework to organize the processes inherent in creating educational activities, thereby ensuring the quality and efficacy of distance learning (Raposo-Rivas; Ibáñez-Cubillas, 2020).

The ADDIE model—an acronym for Analysis, Design, Development, Implementation, and Evaluation—was established in 1975 by the Center for Educational Technology at Florida State University, initially tailored for the U.S. military. However, it was not until 1978, following the work of Dick and Carey, that the model emerged as a benchmark for educational and training program design (Hadi *et al.*, 2017).

Rooted in human knowledge systems theory and the information processing paradigm, the ADDIE model ensures that the outputs of each phase are evaluated before advancing to the next, fostering continuous evaluation and feedback (Maribe, 2009). The interaction between these elements constitutes a five-stage instructional cycle, each with a distinct purpose (Maribe, 2009; Spatioti; Kazanidis; Pange, 2022):

- Analysis: Identifying the core needs of the target audience and assessing students' prior knowledge and competencies.
- Design: Defining the learning environment and curriculum, including objectives, content, activities, and assessment tools. This phase demands precision and a systematic approach.
- Development: Producing instructional materials and integrating technological resources, specifically for online learning environments.
- Implementation: Deploying the final product within the intended learning environments, involving continuous adjustments to optimize performance.
- Evaluation: Systematically assessing training effectiveness. This includes formative evaluation (process-oriented) and summative evaluation (outcome-oriented).

After evaluating alternative models—such as those proposed by Dick and Carey, Gagné, or the ASSURE model—ADDIE was selected for its capacity to facilitate ICT-mediated autonomous learning and its inherent simplicity, which enhances efficiency in course creation (Maribe, 2009). Furthermore, ADDIE is widely utilized to design, develop, and evaluate distance learning experiences and digital educational content (Trust; Pektas, 2018; Raposo-Rivas; Ibáñez-Cubillas, 2020).

This article details the design and development of an open online course aimed at the pre-service and in-service training of teachers in Computational Thinking (CT) and programming via ScratchJr. The goal is to provide the academic community with a framework for initial and continuing education that promotes meaningful, rigorous, and autonomous learning among educators.

2 Materials And Methods

The objective of this study was to design and develop an open online course on ScratchJr for the pre-service and in-service training of education professionals, utilizing the ADDIE instructional design model. To achieve this, a qualitative approach was adopted through a case study framework. As Stake (2005, p. 11) defines it, a case study involves "the study of the particularity and complexity of a single case, in order to understand its activity in important circumstances". In the context of curriculum development, adhering to the phases of an instructional design model is imperative to ensure that planned activities align with learning objectives and facilitate an effective evaluation of outcomes.

The study sample comprised 48 third-year undergraduate students enrolled in the Bachelor's Degree in Pedagogy at the University of Malaga, all of whom completed the initial questionnaire and participated in the training. The initial cohort included 81.3% (n=39) women and 18.8% (n=9) men, with ages ranging from 20 to 25 years. Upon conclusion of the course, 28 students submitted their final ScratchJr projects and completed the exit questionnaire—both of which were mandatory requirements for course completion and the issuance of a certificate by the @rcaComum Association. Of these 28 participants, 78.57% (n=22) were women and 21.42% (n=6) were men, aged between 20 and 24.

2.1 Analysis

During the Analysis phase, the target audience was identified, and comprehensive assessments were conducted to determine the specific training needs of education professionals regarding Computational Thinking (CT) and programming with ScratchJr. This analysis drew upon longitudinal data from the @rcaComum (Miranda-Pinto, 2009) and Kids Media Lab projects, supplemented by a review of existing literature and the application of engagement scales to both pre-service and in-service educators (Miranda-Pinto, 2019; Miranda-Pinto; Osório, 2019). These instruments were pivotal in mapping participants' prior knowledge and target competencies, as well as identifying systemic barriers to the integration of programming technologies within educational settings.

2.1.1 Data collection and initial Data collection and initial analysis

The @rcaComum project (2005–2009) was an Ibero-American online community for education professionals and researchers, designed to facilitate the exchange of best practices and experiences regarding ICT integration within Early Childhood and Primary Education curricula. During this period, semi-structured interviews and content analysis of community forums (Miranda-Pinto, 2009) were employed to identify key areas of interest and the competencies required for effective technology integration. These data revealed a pronounced demand for training in Computational Thinking (CT) to bolster digital literacy from an early age. While the community achieved significant international reach and representation, its focus was subsequently expanded through the Kids Media Lab project (2016–2018). This successor project integrated these foundational principles into specialized training in CT, programming, educational robotics, and internet safety, directly addressing the needs identified within @rcaComum and the continuous evaluation of its activities (Ibáñez-Cubillas, 2018; Miranda-Pinto, 2012; Miranda-Pinto; Osório, 2010).

2.1.2 Justification of the target audience and training needs

Data gathered across these initiatives, particularly within the Kids Media Lab project, underscore the pressing need for professional development in Computational Thinking (CT) using ScratchJr and Scratch. Qualitative analysis of the questionnaires revealed that most early childhood educators lacked foundational programming skills, which constrained their capacity to integrate these concepts into their pedagogical practice. Furthermore, three critical needs were identified: (1) the development of fundamental coding literacy for novice teachers; (2) training on the curricular integration of ScratchJr and Scratch across early childhood and primary education subject areas; and (3) the establishment of a Community of Practice (CoP) to facilitate the exchange of experiences and bolster teachers' self-efficacy in classroom technology use (Miranda-Pinto; Osório, *et al.*, 2017).

2.1.3 Course design and learning objectives

Based on these findings, the course was designed as a flexible, autonomous, and self-paced program, tailored for both pre-service and in-service educators. Developed under the Laboratório de Tecnologias e Aprendizagem de Programação (KML II) project, this training was structured in a modular format, spanning from the foundational concepts of Computational Thinking (CT) to the execution of practical application projects using ScratchJr and Scratch. The course objectives were specifically formulated to bridge the identified knowledge gaps, focusing on the acquisition of practical competencies to foster the pedagogical integration of programming in early childhood and primary education (Miranda-Pinto; Osório, *et al.*, 2017).

This analysis phase established a robust evidence base that informed the creation of specialized content tailored to teachers' specific requirements. The granular identification of the target audience and their developmental needs enabled the precise calibration of learning objectives, materials, and activities, thereby ensuring constructive alignment with the interests and professional demands of the target group.

2.2 Design

In this phase, the training structure of the ScratchJr learning course is planned, adapting it to the findings of the analysis phase and aligning each component with the identified needs of the target audience. The course is designed as an e-learning program so that teachers and educators, both in initial training and in practice, can access the content in a flexible and self-managed way. This choice responds to the data collected, which indicated time constraints among participants, making it necessary to have a structure that could be adapted to different paces and schedules.

2.2.1 Course structure and content organization

The course duration was established at five days, with a recommended commitment of two hours per day of asynchronous study, though this remains flexible to accommodate the individual interests and learning paces of the participants. To facilitate the development of Computational Thinking

(CT) and programming competencies, the content is structured around several core components: thematic introduction, learning objectives, instructional materials, learning activities, assessment, and supplemental reading. These elements were organized using a planning grid (Table 1), informed by the McGee and Reis (2012) alignment model, to ensure rigorous correspondence between objectives, activities, and assessment criteria.

Table 1. ScratchJr course planning.

COURSE ScratchJr					
Week	Content	Objective	Activities	Resources and materials	Evaluation
Day 1 (2 hours)	ScratchJr App ScratchJr Interface ScratchJr Paint Editor ScratchJr Blocks ScratchJr Community Coding Algorithms	LO 1 LO 2 LO 3 LO 4 LO 8	1st hour See the document Information and install the ScratchJr APP, if possible, on a tablet; Visit the website https://www.scratchjr.organdwatchtheScratchJrintroductory video available at https://www.scratchjr.org/about/info; Explore the resources available in the Learn Menu and the Teach Menu on the https://www.scratchjr.orgwebsite. Sign up to the ScratchJr Connect community http://scratchjr.org/teach/connect 2nd hour Read the text "Taking coding home: analysis of ScratchJr usage in home and school settings". Do the "Algorithms" activity, available in the PDF "Algorithm ScratchJr.pdf".	Mandatory video: ScratchJr video presentation (https://www.scratchjr.org/about/info) Obligatory readings: ▪ Interface guide ▪ Paint Editor Guide ▪ Block Guide ▪ Taking coding home with you: an analysis of the use of ScratchJr in home and school environments. Reference materials or activities: ▪ ScratchJr planning Information ▪ ScratchJr Algorithm Optional reading: ▪ Teaching tools, teachers' rules: exploring the impact of teaching styles on young children's programming knowledge in ScratchJr.	Discussion forum: share ideas, questions and reflections with fellow students on the required reading and solving the algorithms. Send the resolution of the "Algorithms" activity to the email geral@kidsmedialab.pt
Day 2 (2 hours)	Coding ScratchJr App ScratchJr Interface ScratchJr Paint Editor ScratchJr Blocks	LO 1 LO 2 LO 5 LO 6 LO 7 LO 8	1st hour Carry out the Coding Cards activities (Activity 1 and Activity 2). 2nd hour Carry out the activity "Coding Cards" (Activity 3). Start the "Final Project Design with ScratchJr". Make the project design (storyboard) in four 4 pages, in ScratchJr.	Activity materials: Coding Cards activity 1 Coding Cards activity 2 Coding Cards activity 3 Designing the final project with ScratchJr Optional reading: Designing ScratchJr: support for early childhood learning through computer programming	Send the resolution of the "Coding Cards" activities to the email geral@kidsmedialab.pt Discussion forum: Share your ideas and questions about the design and creation of your project with your peers.
Day 3 (2 hours)	ScratchJr App ScratchJr Interface ScratchJr Paint Editor ScratchJr Blocks Coding Algorithms	LO 1 LO 2 LO 7 LO 8	1st hour Continued "Designing the Final Project with ScratchJr". 2nd hour Start creating the game or story in ScratchJr. Spend this hour on the first page, creating the scenario and programming the different actors.	Activity materials: Diseñar el proyecto final con ScratchJr Optional reading: Constructing the ScratchJr programming language in the early childhood classroom. Discussion forum: Share your ideas and questions about the design and creation of your project with your peers.	
Day 4 (2 hours)	ScratchJr App ScratchJr Interface ScratchJr Paint Editor ScratchJr Blocks Coding Algorithms	LO 1 LO 2 LO 7 LO 8	1st hour Continue with the creation of the game or story in ScratchJr. Spend this hour on the 2nd page, creating the scenario and programming the different actors. 2nd hour Continue with the creation of the game or story in ScratchJr. Spend this hour on the 3rd page, creating the scenario and programming the different actors.	Activity materials: Diseñar el proyecto final con ScratchJr Optional reading: Programming with ScratchJr: a review of the first year of user analytics	Discussion forum: Share your ideas and questions about the design and creation of your project with your peers.

Day 5 (2 hours)	ScratchJr App ScratchJr Interface ScratchJr Paint Editor ScratchJr Blocks Coding Algorithms	LO 1 LO 2 LO 7 LO 8	1st hour Continue with the creation of the game or story in ScratchJr. Spend this hour on the 4th page, creating the scenario and programming the different actors. 2nd hour At the end, share your ScratchJr project with your classmates on the course platform and by email.	Activity materials: Designing the final project with ScratchJr Optional reading: Bibliography	Send the final project to the email: geral@kidsmedialab.pt Discussion forum: Share your final project and comment on the projects of your peers and comment on what you liked.

Source: Author's own elaboration.

2.2.2 Learning objectives and specific competencies

The Learning Objectives (LOs) were defined to ensure each represents a practical, transferable skill applicable to educators' professional practice. These objectives follow a scaffolded progression, ranging from initial exploration and mastery of the interface (LO1–LO3) to the independent creation and presentation of an original project (LO7). This sequence aims to cultivate an increasing level of proficiency, moving from foundational familiarization to creative application and reflection on pedagogical practice:

- LO1: Identify the core features of the ScratchJr application, relating them to programming principles and justifying its suitability for early childhood education.
- LO2: Explore the ScratchJr interface to evaluate its affordances and limitations for curricular integration in preschool.
- LO3: Analyze ScratchJr activities integrated into the curriculum by examining exemplars from the official website, ScratchJr Connect, and provided resources.
- LO4: Register and engage with the ScratchJr Connect community.
- LO5: Articulate and describe programmed actions within the ScratchJr environment.
- LO6: Execute and adapt Coding Card activities within the platform.
- LO7: Design and develop a comprehensive original project in ScratchJr.
- LO8: Collaborate with peers to discuss experiences, resolve queries, and share insights regarding programming.

2.2.3 Evaluation and reflection

To reinforce learning, the course incorporates formative assessment activities where participants receive feedback from both tutors and peers, fostering a culture of reflective and collaborative evaluation. In the final stage, the exchange of experiences is prioritized, so, participants share their capstone projects within the ScratchJr Connect community and engage in dialogue regarding their challenges and achievements. This interaction not only facilitates collective knowledge construction but also fosters the development of a professional Community of Practice (CoP) dedicated to the integration of ScratchJr in early childhood education.

2.3 Development

During the Development phase, the ScratchJr course is instantiated on the Moodle learning platform within the @rcaComum community. This stage is primarily operational, as it involves the production, structuring, and organization of instructional materials and the Virtual Learning Environment (VLE). These actions are strictly informed by findings from the analysis and design phases to ensure the pedagogical and functional coherence of the course.

2.3.1 Production of teaching materials

Guided by the learning objectives and the structural blueprint established during the design phase, instructional materials were developed to include readings, guides, tutorials, and interactive ScratchJr exemplars. Each resource was designed to be clear, visually engaging, and practice-oriented, fostering autonomous learning while remaining adaptable to the diverse needs and contexts of the participants. Key resources include:

- Theoretical Readings: open-access scientific literature was provided in PDF format to ensure equitable access to core theoretical content.
- Step-by-Step Guides: instructional documents with detailed procedural steps were created to empower participants to navigate ScratchJr independently. These guides feature visual aids and annotated screenshots illustrating each phase of tool utilization.
- Video Tutorials and Interactive Exemplars: fundamental concepts—such as project creation and the application of Coding Cards—were addressed through video tutorials hosted on the official ScratchJr Connect platform, providing dynamic demonstrations of the software in action.

2.3.2 Design of the virtual learning environment

The Moodle platform was configured to provide an intuitive and streamlined environment, facilitating seamless access to instructional materials and enabling participants to progress through the curriculum in a clear, structured manner. The course was organized into thematic modules that align with the predefined learning objectives, designed to guide participants from foundational concepts toward the development of their final projects. Figures 2, 3, 4 illustrate the architectural layout and the technological resources integrated into the ScratchJr course.

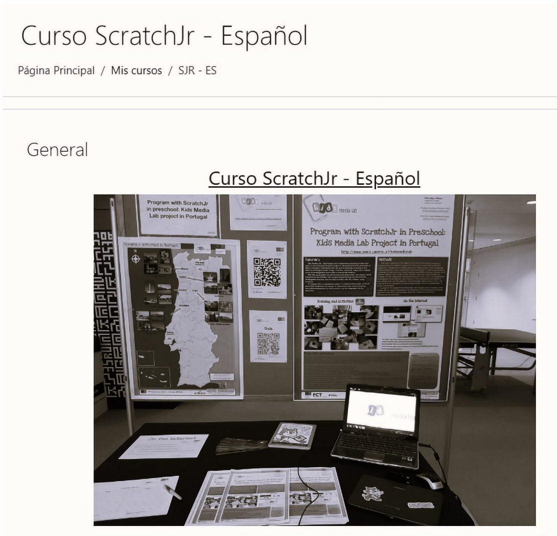

Figure 2. Structure of the ScratchJr course—header, first section, and second section—.
Source: Author's own elaboration.

2.4 Implementation

The Implementation phase of the ScratchJr course was executed in two distinct stages, facilitating an iterative process of continuous improvement based on participant feedback and evaluation data. These stages comprised: (1) a pilot test to refine the initial design, and (2) the case study, which serves as the focal point of this article.

2.4.1 Design of the virtual learning environment

During the first stage, the ScratchJr course was launched as a pilot program within the Early Childhood Technology Certificate Program at Tufts University (USA) during the 2020/2021 academic year. This initial implementation, conducted in July 2021, targeted a cohort of 25 participants enrolled in the Special Topics: Early Childhood Technology Residency course (Miranda-Pinto, 2022). The group consisted of education professionals and students specializing in early childhood technology, specifically focusing on the pedagogical application of ScratchJr for young learners.

The objective of this pilot phase was to conduct a comprehensive evaluation of the course, assessing content relevance, instructional clarity, platform navigability, and the efficacy of activities in achieving learning objectives. To this end, participants completed a post-course evaluation questionnaire,

Figure 3. Structure of the ScratchJr course—header, first section, and second section—.
Source: Author's own elaboration.

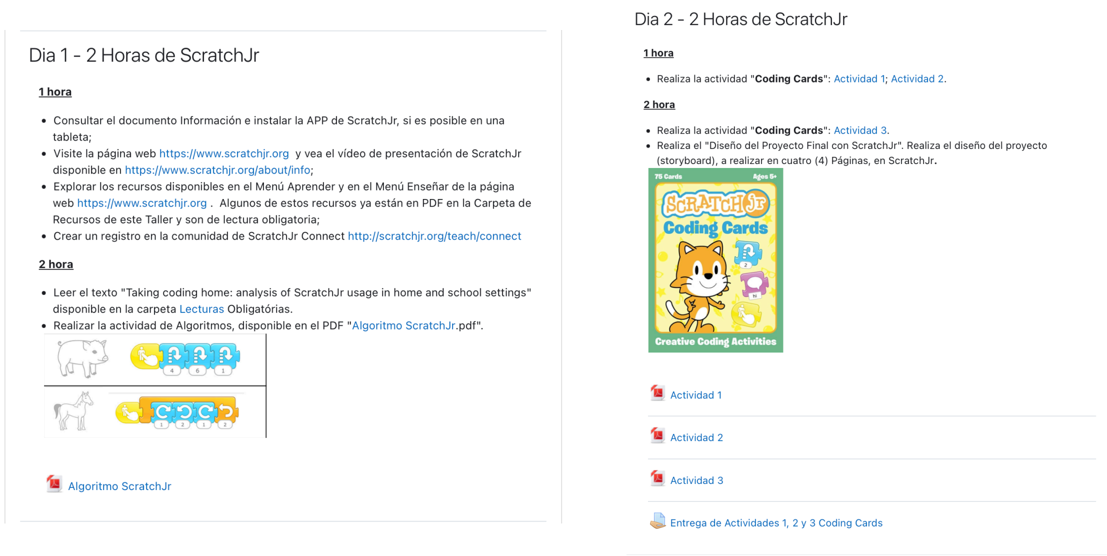

Figure 4. Daily structure of the ScratchJr course.
Source: Author's own elaboration.

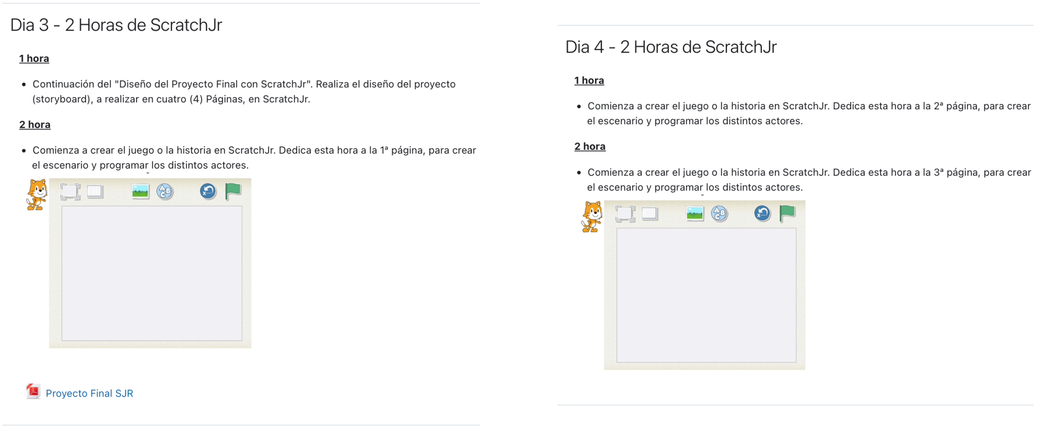

Figure 5. Daily structure of the ScratchJr course.
Source: Author's own elaboration.

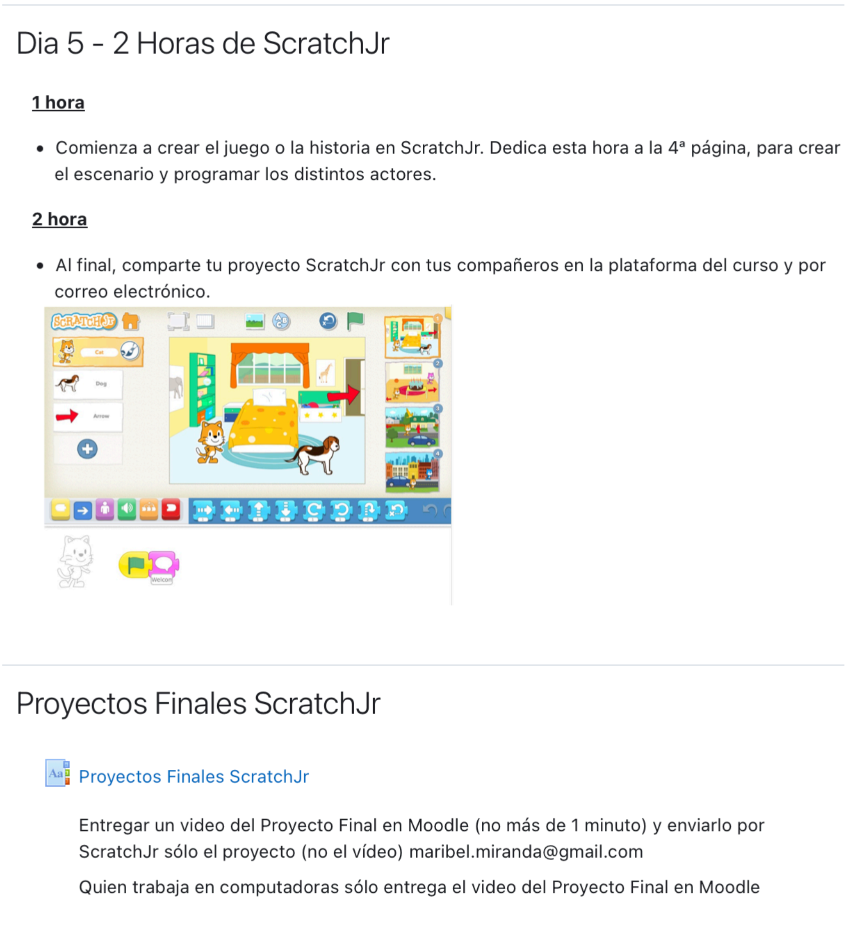

Figure 6. Daily structure of the ScratchJr course.
Source: Author's own elaboration.

providing both qualitative and quantitative data. Key metrics included the clarity of instructional materials, the usability and organization of the Virtual Learning Environment (VLE), overall satisfaction with practical tasks, and the perceived pedagogical utility of the knowledge acquired (Miranda-Pinto, 2022).

Based on these findings, significant refinements were made to the course design. These improvements included restructuring activities, incorporating supplementary instructional resources, and streamlining the Moodle interface to enhance navigation and module sequencing. These modifications aimed to mitigate identified challenges and enrich the autonomous learning experience, ensuring the course was both accessible and effective for pre-service and in-service teachers.

2.4.2 Stage 2. Case study

Incorporating the refinements previously outlined, the ScratchJr course was offered to undergraduate students in the Degree in Education at the University of Malaga (Spain), as a specialized component of the curriculum. A total of 48 students enrolled in and completed the course; of these, 28 submitted a final project along with pre- and post-test questionnaires, fulfilling the requirements for the certificate of participation issued by the @rcaComum Association.

Throughout this implementation phase, participants were monitored to evaluate the course's impact on their development of Computational Thinking (CT) and programming competencies. The final project required the creation of an original story or game in ScratchJr, enabling students to apply acquired knowledge within a self-directed creative context. Furthermore, a moderated discussion forum was established on the platform to facilitate experience sharing, peer-to-peer support, and collaborative problem-solving, thereby fostering an active learning community.

To assess the efficacy of the implementation and gather empirical data on learning outcomes, students completed a comprehensive post-test questionnaire featuring both open-ended and closed-ended items. The results (detailed in Ibáñez-Cubillas, Miranda-Pinto, and López-Rodríguez (2023)) indicated that participants perceived the course as highly valuable and relevant to their future professional careers. Notably, a significant increase was observed in students' self-efficacy regarding the integration of programming tools like ScratchJr into their future educational practices.

2.5 Evaluation

Within the ADDIE framework, the Evaluation phase is critical to verifying the quality and overall efficacy of the training program. In this study, a dual evaluation approach was adopted: (1) a formative evaluation, centered on the attainment of learning objectives and the immediate impact on participants' competencies; and (2) a summative course evaluation, designed to identify necessary refinements to the content, delivery, and structural framework for future iterations.

2.5.1 Formative Assessment

The formative assessment sought to measure the degree of content mastery and skill acquisition upon course completion, thereby determining the extent to which learning objectives were met. Prior to the intervention, students (n=48) completed a baseline questionnaire regarding general knowledge of Programming Languages in Education. This instrument comprised three sections: (1) introduction; (2) demographic data (gender, age, university, and degree); and (3) specialized knowledge, consisting of seven items five dichotomous, one multiple-choice with an open-entry option, and one qualitative open-ended question.

Subsequently, students engaged with the ScratchJr online course, completing modular assessment activities and submitting a capstone project (as detailed in Table 1). While all 48 students completed the instructional phase, 28 submitted the final project required for certification. Upon completion, these participants (n=28) answered a summative questionnaire consisting of nine items designed to evaluate: (a) mastery of fundamental concepts (programming languages, Computational Thinking, and coding); (b) the perceived impact of the training on their future professional practice; and (c) the competencies acquired for cross-curricular ScratchJr integration.

This final instrument utilized a mix of open-ended, dichotomous, and multi-option Likert-style scales. Results (Ibáñez-Cubillas; Miranda-Pinto; López-Rodríguez, 2023) revealed that, prior to the course, students possessed limited foundational knowledge of programming languages (77.1%), ScratchJr (79.2%), and Computational Thinking (87.5%). Notably, 95.8% were unfamiliar with other child-oriented programming languages, and 97.9% reported no prior programming proficiency. Following the intervention, participants demonstrated a significant acquisition of coding and CT competencies. Furthermore, between 96.4% and 100% of students affirmed the necessity of incorporating CT and ScratchJr into initial teacher education, citing its profound relevance to their prospective professional careers (Ibáñez-Cubillas; Miranda-Pinto; López-Rodríguez, 2023).

2.5.2 Course evaluation

Upon completion of the course, students filled out a questionnaire to evaluate the suitability and quality of the course structure, content, and instructional activities (Miranda-Pinto; Ibáñez-Cubillas; López-Rodríguez, 2023). This instrument comprised seven items, five of which were dichotomous (yes/no) to assess the relevance, clarity, and adequacy of the provided documents, PDF readings, and learning tasks. Each dichotomous item was paired with an open-ended qualitative field for improvement suggestions, alongside a specific question regarding the overall course framework. Finally, participants provided a global rating of the course using a 5-point Likert scale (where 1 indicates the lowest and 5 the highest level of satisfaction).

The results indicated that 100% of the students perceived the documentation as well-structured, providing clear information regarding organization, required resources, and activities. Regarding content, 100% of participants rated the readings as adequate and sufficient for mastering ScratchJr fundamentals. Furthermore, 92.9% stated that the "Coding Cards" activities were effective for learning core programming concepts, while 96.4% considered the Storyboard design activity to be a critical preliminary step for the successful execution of the final ScratchJr project. The course achieved an overall satisfaction rating of 4.60 out of 5 (Miranda-Pinto; Ibáñez-Cubillas; López-Rodríguez, 2023).

3 Discussion

This study has demonstrated the efficacy of the ADDIE instructional model as a robust structural framework for designing and developing a virtual, self-paced course focused on ScratchJr. While other prominent models such as those by Dick, Carey, and Carey (2009) and Gagné *et al.* (2005), or ASSURE (Heinich *et al.*, 2002) offer valuable instructional design frameworks, ADDIE distinguishes itself through its practicality and systematic flexibility. This characteristic is particularly advantageous for virtual training environments, as it facilitates the precise alignment of learning objectives, activities, resources, and assessment criteria (Hadi *et al.*, 2017).

The findings reflect a significant enhancement in participants' Computational Thinking (CT) competencies and their proficiency in utilizing ScratchJr as a pedagogical tool. These results align with previous research highlighting the pervasive lack of CT training in Initial Teacher Education (ITE) programs (Bers; Sullivan, 2018), while simultaneously validating the power of well-structured interventions to bridge this gap. The scaffolded integration of resources such as Coding Cards and storyboarding facilitated the progressive mastery of programming concepts. Combined with high participant satisfaction, these outcomes underscore the effectiveness of the adopted pedagogical design.

Furthermore, the course catalyzed a shift in students' perceptions of educational programming. With approval ratings between 96.4% and 100%, participants recognized the necessity of integrating these competencies into their training, signaling a heightened motivation to leverage ScratchJr in their prospective professional practice. This is particularly significant given that educators are the primary agents of innovation and digital literacy from the earliest stages of schooling (Bers; Flannery, *et al.*, 2014).

Finally, the institutional backing of the Scratch Education Collaborative, alongside its synergy with international projects such as @rcaComum, Kids Media Lab, and KML II (responsible for the MOOC "Programming and Robotics in Primary and Preschool Education" in Portugal), reinforces the sustainability and scalability of this proposal. Such international collaborations facilitate the integration of CT into the curriculum through a multilingual, multicultural, and inclusive lens, resonating with the Coding as Another Language (CAL) curriculum (Bers, 2022b).

4 Conclusion

The findings of this study support the conclusion that the ADDIE model is a highly effective framework for designing virtual courses centered on educational programming with ScratchJr. Its clear, methodical, and adaptable structure facilitates the strategic planning and execution of training programs that integrate the essential knowledge, skills, and attitudes required for teaching in digital contexts. While other established models such as those by Dick, Carey, and Carey (2009) and Gagné *et al.* (2005), or ASSURE (Heinich *et al.*, 2002) provide valuable guidance, ADDIE distinguishes itself through its pragmatic applicability, particularly within technology-mediated instructional environments.

The adopted e-learning modality has proven to be a viable pathway for delivering high-quality, accessible, and flexible training. It effectively addresses geographical and economic barriers, as well as the educational complexities of the post-pandemic era. Beyond fostering critical digital competencies, this proposal has catalyzed a shift in the attitudes of pre-service educators toward the pedagogical use of programming, increasing their self-efficacy in integrating Computational Thinking into their professional practice.

The positive impact of the course evidenced by the measurable growth in foundational knowledge and high participant satisfaction underscores the necessity of embedding programming and CT into both Initial Teacher Education (ITE) and continuing professional development curricula. This experience reinforces the importance of evidence-based instructional design, utilizing scaffolded activities and specialized resources such as Coding Cards and storyboarding.

Furthermore, the international scope of the course, endorsed by the Scratch Education Collaborative (SEC) and integrated into initiatives like the Portuguese MOOC "Programação e Robótica na Educação Básica e Pré-escolar" (Amante; Souza; Quintas-Mendes; Miranda-Pinto, 2023), demonstrates its global relevance and transferability across diverse cultural and educational landscapes.

In conclusion, this proposal stands as a replicable and significant training resource that contributes to narrowing the digital divide and strengthening digital literacy from early childhood. Future research should prioritize longitudinal monitoring of the impact of such training on classroom practice and explore adaptations for various educational levels and sociocultural realities, thereby consolidating the role of teacher education in addressing the challenges of the 21st century.

References

AMANTE, Lúcia; SOUZA, Elizabeth Batista de; QUINTAS-MENDES, António; MIRANDA-PINTO, Maribel. Designing a MOOC on Computational Thinking, Programming and Robotics for Early Childhood Educators and Primary School Teachers: A Pilot Test Evaluation. *Education Sciences*, v. 13, n. 9, p. 863, 2023. DOI: 10.3390/educsci13090863.

AMANTE, Lúcia; SOUZA, Elizabeth Batista de; QUINTAS-MENDES, António; MONTEIRO, Ana Francisca; MIRANDA-PINTO, Maribel; OSÓRIO, António; ARAÚJO, Catarina. Computational Thinking, Programming and Robotics in Basic Education: Evaluation of an In-Service Teacher's Training B-Learning Experience. *In:* ICERI2019 Proceedings. [*S. l.: s. n.*], 2019. v. 1, p. 10698–10705. DOI: 10.21125/iceri.2019.2626.

BERS, Marina Umaschi. Positive Technological Development: Working with Computers, Children and the Internet. *MassPsych*, v. 51, n. 1, p. 1–6, 2007. Available from: http://ase.tufts.edu/devtech/publications/masspsych.pdf.

BERS, Marina Umaschi. *Blocks to Robots: Learning with Technology in the Early Childhood Classroom*. [*S. l.*]: Teachers College Press, 2008.

BERS, Marina Umaschi. *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. [*S. l.*]: Routledge, 2017. DOI: 10.4324/9781315398945.

BERS, Marina Umaschi. *Beyond Coding: How Children Learn Human Values through Programming*. [*S. l.*]: The MIT Press, 2022. DOI: 10.7551/mitpress/13775.001.0001.

BERS, Marina Umaschi. *Coding As Another Language (CAL) for ScratchJr (CAL-ScratchJr)*. [*S. l.: s. n.*], 2022. DevTech Research Group. Available from: https://sites.bc.edu/codingasanotherlanguage/.

BERS, Marina Umaschi; FLANNERY, Louise; KAZAKOFF, Elizabeth R.; SULLIVAN, Amanda. Computational Thinking and Tinkering: Exploration of an Early Childhood Robotics Curriculum. *Computers & Education*, v. 72, p. 145–157, 2014. DOI: 10.1016/j.compedu.2013.10.020.

BERS, Marina Umaschi; RESNICK, Mitchel. *The Official ScratchJr Book: Help Your Kids Learn to Code*. [*S. l.*]: No Starch Press, 2016. ISBN 9781593276713.

BERS, Marina Umaschi; SULLIVAN, Amanda. *ScratchJr Coding Cards: Creative Coding Activities*. [*S. l.*]: No Starch Press, 2018. ISBN 9781593278991.

BRENNAN, Karen; RESNICK, Mitchel. New Frameworks for Studying and Assessing the Development of Computational Thinking. *In:* PROCEEDINGS of the 2012 Annual Meeting of the American Educational Research Association. Vancouver, Canada: [*s. n.*], 2012. Available from: http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf.

DICK, Walter; CAREY, Lou; CAREY, James O. *The Systematic Design of Instruction*. 7. ed. Upper Saddle River, NJ: Merrill/Pearson, 2009. ISBN 9780205585564.

ESPADEIRO, Rui Gomes. O Pensamento Computacional no currículo de Matemática. *Educação e Matemática*, n. 162, p. 5–10, 2021. Available from: https://em.apm.pt/index.php/em/article/view/2737.

GAGNÉ, Robert M.; WAGER, Walter W.; GOLAS, Katharine C.; KELLER, John M. *Principles of Instructional Design*. 5. ed. [*S. l.*]: Wadsworth, 2005.

HADI, S. P. I.; KUNTJORO, T.; SUMARNI, S.; ANWAR, M. C.; WIDYAWATI, M. N.; PUJIASTUTI, R. S. E. The Development of E-Partograph Module as A Learning Platform for Midwifery Students: The ADDIE Model. *Belitung Nursing Journal*, v. 3, n. 2, p. 148–156, 2017. DOI: 10.33546/bnj.77.

HEINICH, Robert; MOLENDA, Michael; RUSSELL, James D.; SMALDINO, Sharon E. *Instructional Media and Technology for Learning*. 6. ed. [*S. l.*]: Pearson Education Ltd., 2002.

IBÁÑEZ-CUBILLAS, Pilar. *Aspectos educativos y formativos de la atención infantil temprana en comunidades virtuales*. 2018. Tesis de doctorado – Universidad de Granada. Available from: http://hdl.handle.net/10481/52502.

IBÁÑEZ-CUBILLAS, Pilar; MIRANDA-PINTO, Manuela; LÓPEZ-RODRÍGUEZ, Sergio. Evaluation of the Impact of Training in Programming with ScratchJr on Future Pedagogy Professionals. *In:* EDULEARN23 Proceedings. [*S. l.*]: IATED, 2023. p. 6900–6904. DOI: 10.21125/edulearn.2023.1815.

MARIBE, Robert. *Instructional Design: The ADDIE Approach*. [*S. l.*]: Springer, 2009. DOI: 10.1007/978-0-387-09506-6.

MCGEE, Patricia; REIS, Amanda. Blended Course Design: A Synthesis of Best Practices. *Journal of Asynchronous Learning Networks*, v. 16, n. 4, p. 7–22, 2012. Available from: https://www.learntechlib.org/p/89268/.

MIRANDA-PINTO, Maria Susana. *Processos de Colaboração e Liderança em Comunidades de Prática Online - O Caso da @rcaComum, uma Comunidade Ibero-Americana de Profissionais de Educação de Infância*. 2009. Tese de doutoramento – Instituto de Estudos da Criança. Available from: http://repositorium.sdum.uminho.pt/bitstream/1822/12571/4/Tese_Doutoramento_20-18-02-09.pdf.

MIRANDA-PINTO, Maria Susana. Ambientes emergentes para o desenvolvimento de comunidades de práctica online – a @rcaComum. *In:* DIAS, Paulo; OSÓRIO, José (eds.). *TIC na educação: Perspectivas de Inovação*. [*S. l.*]: Centro de Competências da Universidade do Minho, 2012. p. 46–64.

MIRANDA-PINTO, Maria Susana. Programación y robótica en educación infantil: Estudio multicaso en Portugal. *Prisma Social*, n. 25, p. 248–276, 2019. Available from: https://revistaprismasocial.es/article/view/2733.

MIRANDA-PINTO, Maria Susana. ScratchJr and Scratch Online Training for Early Childhood Educators and Primary School Teachers. *In:* GÓMEZ CHOVA, Luis; LÓPEZ MARTÍNEZ, Agustín (eds.). *INTED2022 Proceedings*. [*S. l.*]: IATED Academy, 2022. p. 821–830. DOI: 10.21125/inted.2022.

MIRANDA-PINTO, Maria Susana; IBÁÑEZ-CUBILLAS, Pilar; LÓPEZ-RODRÍGUEZ, Sergio. Evaluation of the Design of an Online Course for ScratchJr Learning in Students of the Pedagogy Undergraduate Degree. *In:* EDULEARN23 Proceedings. [*S. l.*]: IATED, 2023. p. 6895–6899. DOI: 10.21125/edulearn.2023.1815.

MIRANDA-PINTO, Maria Susana; OSÓRIO, António. Colaboración y aprendizaje en el ciberespacio. @rcacomum: comunidad de práctica iberoamericana de educadores de infancia. *Revista Mexicana de Investigación Educativa*, v. 15, n. 44, p. 35–64, 2010. Available from: https://www.scielo.org.mx/pdf/rmie/v15n44/v15n44a4.pdf.

MIRANDA-PINTO, Maria Susana; OSÓRIO, António. Aprender a programar en Educación Infantil: análisis con la escala de participación. *Pixel-Bit, Revista de Medios y Educación*, n. 55, p. 133–156, 2019. DOI: 10.12795/pixelbit.2019.i55.08.

MIRANDA-PINTO, Maria Susana; OSÓRIO, António. ScratchJr App in Portuguese Schools: Kids Media Lab Project. *In:* ICERI2020 Proceedings. [*S. l.: s. n.*], 2020. v. 1, p. 5709–5718. DOI: 10.21125/iceri.2020.1227.

MIRANDA-PINTO, Maria Susana; OSÓRIO, António; MONTEIRO, Ana Francisca; VALENTE, Luís; ARAÚJO, Catarina. Laboratory of Technologies and Learning of Programming and Robotics for Pre and Primary School. *In:* ICERI2017 Proceedings. [*S. l.: s. n.*], 2017. v. 1, p. 1497–1502. DOI: 10.21125/iceri.2017.0473.

PAPERT, Seymour. *Logo: Computadores e educação*. [*S. l.*]: Brasiliense, 1985.

QUINTAS-MENDES, António; SOUZA, Elizabeth Batista de; AMANTE, Lúcia. Learning Design of a Programming and Robotics MOOC for Childhood Teachers and Educators. *Prisma Social*, n. 39, p. 234–261, 2022. Available from: https://revistaprismasocial.es/article/view/4807.

RAPOSO-RIVAS, Manuela; IBÁÑEZ-CUBILLAS, Pilar. Innovación educativa con tecnologías. *In:* RAPOSO-RIVAS, Manuela; CEBRIÁN DE LA SERNA, Manuel (eds.). *Tecnologías para la formación de educadores en la sociedad del conocimiento*. [*S. l.*]: Pirámide, 2020. p. 203–214.

RESNICK, Mitchel; ROSENBAUM, Eric. Designing for Tinkerability. *In:* HONEY, Margaret; KANTER, David E. (eds.). *Design, Make, Play: Growing the Next Generation of STEM Innovators*. [*S. l.: s. n.*], 2013. p. 163–181. Available from: http://web.media.mit.edu/~mres/papers/designing-for-tinkerability.pdf.

ROQUE, Ricarose. *A Facilitator's Guide to... Family Creative Learning: The ScratchJr Edition*. [*S. l.: s. n.*], July 2022. Family Creative Learning. Available from: https://www.familycreativelearning.org/guide/.

ROQUE, Ricarose; LEGGETT, Sarah. *Family Creative Learning Facilitator Guide*. [*S. l.: s. n.*], Aug. 2014. Family Creative Learning. Available from: http://familycreativelearning.org/guide/.

SOUZA, Elizabeth Batista de; AMANTE, Lúcia; QUINTAS-MENDES, António. Evaluation of a B-Learning Training for Teachers in Computational Thinking, Programming and Robotics: Preparing the Design of a MOOC. *In:* ICERI2020 Proceedings. [*S. l.: s. n.*], 2020. v. 1, p. 1655–1664. DOI: 10.21125/iceri.2020.0418.

SPATIOTI, A. G.; KAZANIDIS, I.; PANGE, J. A Comparative Study of the ADDIE Instructional Design Model in Distance Education. *Information*, v. 13, n. 9, p. 402, 2022. DOI: 10.3390/info13090402.

STAKE, Robert E. *Investigación con estudio de casos*. Madrid: Morata, 2005.

TRUST, Torrey; PEKTAS, Ebru. Using the ADDIE Model and Universal Design for Learning Principles to Develop an Open Online Course for Teacher Professional Development. *Journal of Digital Learning in Teacher Education*, v. 34, p. 219–233, 2018. DOI: 10.1080/21532974.2018.1494521.

WING, Jeannette M. Computational Thinking. *Communications of the ACM*, v. 49, n. 3, p. 33–35, 2006. DOI: 10.1145/1118178.1118215.

WING, Jeannette M. Computational Thinking and Thinking about Computing. *The Royal Society*, v. 366, p. 3717–3725, 2008. DOI: 10.1098/rsta.2008.0118.

Author contributions

Maribel dos Santos Miranda-Pinto: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Visualization, Project administration, Funding acquisition, Writing – original draft, Writing – review and editing; **Pilar Ibáñez-Cubillas**: Conceptualization, Methodology, Validation, Investigation, Resources, Data curation, Visualization, Project administration, Writing – original draft, Writing – review and editing; **Slava López-Rodríguez**: Validation, Resources, Writing – original draft, Writing – review and editing.

Data availability

Research data are available in the body of the document.

Funding

Laboratório de Educação a Distância e eLearning (LE@D), financiado por Fundos Nacionais através da FCT, no âmbito dos projetos UID/4372/2025 e UID/PRR/04372/2025. Arcacomum Associação emPortugal: https://www.arcacomum.pt. Scratch Education Collaborative (SEC): 2022-2024 Cohort.