Project Number: 101006468

Project Acronym: PAFSE

Project title: Partnerships for Science Education

EDUCATIONAL SCENARIOS
UMINHO UNIVERSITY (UM)

Universidade do Minho

JULY 2023

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

2. Specifications for an educational scenario on the topic of "Low-code development environments – Level 2 (Intermediate)"

Title of Educational Scenario: Connecting students to IT using low-code development environments to promote public health and digital literacy – Level 2 (Intermediate)
Topic in School Curriculum: Block programming / Health subject by teachers' choice
School Subject: ICT classes/Biology/F.Q classes/Health Education classes (Interdisciplinarity**1**)
Main resource: MIT App Inventor
Grade level: 8th grade (+/- 13-14 years old students)

Context and its relevance to public health education

The technological revolution of the last decades has contributed to the consolidation of a new social paradigm known as knowledge society or information society. This paradigm is reflected in a globalized and multilingual world, full of economic, commercial, political, social, and cultural relations, where professional specialization is a necessity. Aiming to help achieve this specialization, the educational scenario supports ICT teachers in expanding students' skills in a way they are not just passive consumers of technology, but active content creators too. Learning how to code can support students' engagement in the development of innovative solutions that benefit the health of their community, while developing general problem-solving skills central to success in STEM (Science, Technology, Engineering, Mathematics) curricula and careers. By learning how to code students go from being passive users of apps, digital content, and web pages to actively participate in their creation with meaningful purpose.
Particularly, block-based coding or programming is an element of programming where text-based computer commands are groups together in pre-programmed blocks that drag and drop together to build computer programs such as animations and games. Block coding is considered "syntax-free" in that a user does not need to be careful about the order and requisite syntax of commands and punctuation, which need to be memorized in text-based programming. This means it has a tremendous potential to take education to the next level.
The scenario aims to familiarize students with public health risks and patterns of protective behavior, as well as making them capable of explaining those ideas to others in low-code environments. Several topics related to a main determinant of health - nutrition habits - will be explored while operating in various multimodal content creation tools. The learning experience supports youths in understanding how STEM may contribute to create new and revolutionizing solutions to public health, as well as stimulate their creativity, decision-making and problem-

1 Integrating knowledge and methods from different disciplines, using a real synthesis of approaches.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

solving skills, while supporting them in the process of becoming tech producers and public health ambassadors.

Estimated Duration

Variable, depending on ICT teachers weekly schedule (normally 40 min. per week)
Estimate (based on pilot experience): At least 4 classes of 40 min. (lesson 1- 4) and 3 sessions of 40-45 minutes for supplementary learning activities and school project (session 4 – session 7)

Classroom organization requirements

Classroom ergonomics:

- Create a space that is adaptable to the learning experience instead of having the learning experience adapted to the space (Bayse, 2015);
- Focused on the training of required skills and collaborative work;
- Teachers are practically merely content curators and learning facilitators – Students are required to be as autonomous as possible.
- Teachers are required to provide support to students, without compromising students autonomy.

For the learning-through-teacher lessons, students will work alone/in groups and should have access to:

- An ICT classroom with regular functioning computers;
    - Setup - MIT App Inventor;
        - System Requirements - MIT App Inventor;
        - App tester - MIT App Inventor;
        - Pre-setup (Tech and Networking Specialists) - MIT App Inventor.
- An internet connection;
- A gmail account (to log in in MIT App Inventor);
    - Accounts and devices - MIT App Inventor.
- Any android device.

To carry out the research project, students will work in groups and the same equipment is required, as well as an open, curious, and creative mind.

Observations:

- No prior downloading of software is required;

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006468.

- Students are welcome to use their own computers;
- Each student should have their own email account;
- App Inventor offers the ability to develop using the Android emulator that shows up in a window on the computer screen if the students don't have an android device. However, using the emulator isn't as good as a physical device, because students can't carry their apps around with them and some features might not be present;
- The navigator "Internet Explorer" is not supported;
- MIT App Inventor works as a cloud, therefore everything is stored online.

Prerequisite knowledge and skills

- Basic IT and ICT notions.

General content glossary

- **IT**. **IT** (Information Technology) is the study, design, development, application, implementation, support, or management of computer-based information systems. (Source: Code Academy)
- **ICT**. Information and communication technologies (**ICT**) is defined as a diverse set of technological tools and resources used to transmit, store, create, share or exchange information. These technological tools and resources include computers, the Internet (websites, blogs and emails), live broadcasting technologies (radio, television and webcasting), recorded broadcasting technologies (podcasting, audio and video players, and storage devices) and telephony (fixed or mobile, satellite, visio/videoconferencing, etc.). (Source: UNESCO)
- **Low-code.** A **low-code** platform allows app development through the use of a graphical user interface (GUI) rather than traditional hand-coding. In other words, it is a type of visual software development environment that allows developers to drag and drop application components, connect them together and create mobile or web apps with little to no code. (Source: Techtarget)
- **Block coding**. **Block coding** is a process used in computer programming where text-based software codes change to a visual block format to create animated games, characters, and even stories. With block coding, kids can learn the basics and foundational concepts through visuals instead of text-based coding. (Source: Codingal)
- **Algorithm.** An **algorithm** is a detailed step-by-step instruction set or formula for solving a problem or completing a task. In computing, programmers write algorithms that instruct the computer how to perform a task. When you think of an algorithm in the most general way (not just in regards to computing), algorithms are everywhere. A recipe for making food is an algorithm, the method you use to solve addition or long

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

division problems is an algorithm, and the process of folding a shirt or a pair of pants is an algorithm. (Source: Tynker - Coding for Kids)

- **Programming language**. A **programming language** is a set of commands, instructions, and other syntax use to create a software program. In other words, it is a language that allows a programmer to tell the computer what to do in a variety of circumstances. Languages that programmers use to write code are called "high-level languages." This code can be compiled into a "low-level language," which is recognized directly by the computer hardware. (Source: Techterms; Ageuk)
- **Event-driven programming**. **Event-driven programming** is a programming paradigm in which the flow of program execution is determined by *events* - for example a user action such as a mouse click, key press, or a message from the operating system or another program. An event-driven application is designed to detect events as they occur, and then deal with them using an appropriate *event-handling procedure*. (Source: Technologyuk)
- **MIT App Inventor**. **MIT App Inventor** is an intuitive, visual programming environment that allows everyone – even children – to build fully functional apps for Android phones, iPhones, and Android/iOS tablets. It is an open-source tool that aims to make programming and app building accessible to a wide variety of audiences (educators; researchers; government; etc.) Initially developed by Professor Hal Abelson and his team, App Inventor is managed by members of MIT's Center for Mobile Learning. (Source: MIT App Inventor)
- **IDE**. An **IDE**, or Integrated Development Environment, enables programmers to consolidate the different aspects of writing a computer program and develop programs more efficiently. IDEs increase programmer productivity by combining common activities of writing software into a single application: editing source code, building executables, and debugging. (Source: Code Academy)
- **User Interface.** The **user interface** (UI) is the look and feel of an operating system. A good interface puts the user first, making commands and access to apps easy to discover. For the programmer, understanding how the interface works and what impact it has on application design is extremely useful. (Source: O'Reilly)
- **Conditional blocks.** Conditionals refer to expressions or statements that evaluate to true or false. If the condition is "true", a particular section of text will be inserted into the message. If the condition is "false", the text will not be inserted. An "ELSE" clause can be included as part of the conditional statement so that a different section of text will be inserted into the message when the condition is "false". (Source: Isoft)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

- **Loops** are a way to tell a computer to do something many times in a row. Computers are really good at doing things over and over again, and doing them fast. (Source: technovationchallenge)
- **Lists** - a way to organize multiple pieces of data in App Inventor (Source: technovationchallenge)
- **Index** - a number that tells you where a piece of data is in a list (Source: technovationchallenge)
- **Variable**. A **variable** is a container that holds a single number, word, or other information that you can use throughout a program. A variable is like a chest you can fill with different values. Component properties are variables that are built into a component. Event parameters are special variables that give you extra information about an event. Global variables have global scope, meaning that they can be set and read from any blocks in the workspace. Local variables have local scope, meaning that they exist only within their initialization block, which has space to add more blocks. (Source: Idtech; O'Reilly)
- **Procedure.** A **procedure** is a set of instructions that is grouped together, given a name, and made available for later use. This makes your code easier to read, think about, and change. Ultimately, using a procedure is more powerful. The steps for getting started are straightforward. (Source: O'Reilly)

Pedagogical glossary

- **Constructivism.** Jean Piaget presented the theory of **constructivism**, asserting that knowledge is not simply transmitted from teacher to student, but actively constructed in the mind of the learner. Learners don't receive ideas; rather they create them from their own base of knowledge. Some characteristics of constructivist learning are that it:
  - ⇒ fosters critical thinking;
  - ⇒ creates motivated and independent learners;
  - ⇒ has lessons that include guided discovery, whereby the teachers acts as a guide to the learner, helping to point out inconsistencies in students' thinking. Students build their understanding by resolving these conflicts;
  - ⇒ includes a minimal amount of direct instruction. (Source: MIT App Inventor)
- **Constructionism.** Building from the idea of **constructivism**, Seymour Papert presented his theory of constructionism which suggests that new ideas are most likely to be created when learners are actively engaged in building some type of external artifact that they can reflect upon and share with others. Elements of a constructionist learning environment include:
  - ⇒ a teacher who acts as a facilitator;
  - ⇒ learners who investigate, create, and solve problems;
  - ⇒ learner collaboration;

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

- ⇒ learners engaging in authentic tasks;
- ⇒ opportunity for feedback and multiple opportunities for revision. (Source: MIT App Inventor)

- **Problem-Based Learning. Problem-based learning** is one type of constructivist learning theory that can be applied in a classroom setting. It is a method which allows students to learn about a subject by exposing them to multiple problems, so they will be able to construct their understanding of the subject through these problems. Problem-based learning typically:
  - ⇒ begins with problem for students to solve or learn about;
  - ⇒ includes problems that are somewhat ambiguous to mirror the complexity of real life;
  - ⇒ uses an inquiry model;
  - ⇒ requires students to present a conclusion of the problem solving process, but does not necessarily require them to create a product as a result;
  - ⇒ is driven by defined problems. (Source: MIT App Inventor)
- **Project-Based Learning. Project-based learning** encompasses Papert's theory of constructionism where students build an artifact as part of the learning process. Project-based learning typically:
  - ⇒ begins with an end product in mind;
  - ⇒ includes production of an artifact, which typically raises one or more problems for students to solve;
  - ⇒ asks students to use or present the product they have created;
  - ⇒ is driven by the end product;
  - ⇒ stresses that content knowledge and skills acquired during the production process are critical to success. (Source: MIT App Inventor)
- **Computational thinking**. The term **Computational Thinking** (CT), coined by Jeannette Wing in 2006, describes solving problems, designing systems, and understanding human behavior based on the principles of computer science. CT includes analyzing and organizing data, automated problem solving and using it to solve similar problems. Nowadays, it has become necessary to solve complex technological problems. If sufficient background knowledge is available and the necessary new knowledge is acquired through critical thinking, CT may help to solve the problem. It is actually a hybrid of several other modes of thinking, like abstract, logical, algorithmic, constructive and modelling thinking, which summarizes all previous modes for solving the corresponding problem. (Source: IGI)
- **Brainstorming**. Brainstorming is a group problem-solving technique that involves the spontaneous contribution of ideas from all members of the group. (Source: Merriam-Webster)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

- **Collaborative learning.** A **collaborative** (or cooperative) **learning** approach involves students working together on activities or learning tasks in a group small enough to ensure that everyone participates. Students in the group may work on separate tasks contributing to a common overall outcome or work together on a shared task. This is distinct from unstructured group work. Some collaborative learning approaches put mixed ability pairs, groups or teams together to work in competition with each other in order to drive more effective collaboration. (Source: Evidence For Learning)
- **Gamification**. **Gamification** of education is a developing approach for increasing learners' motivation and engagement by incorporating game design elements in educational environments. It is often described as the use of game design elements in non-game contexts" (Deterding, Dixon, Khaled, & Nacke, 2011), "the phenomenon of creating gameful experiences" (Hamari, Koivisto, & Sarsa, 2014), or "the process of making activities more game-like" (Werbach, 2014). (Source: Springer)
- **Learning through Storytelling**. **Storytelling** is the vivid description of ideas, beliefs, personal experiences, and life-lessons through stories or narratives that evoke powerful emotions and insights. It represents the use of stories or narratives as a communication tool to value, share, and capitalize on the knowledge of individuals. (Source: Springer)
- **STEM**. **STEM** (Science, Technology, Engineering, and Math) is an integrated, interdisciplinary, and student-centered approach to learning that encourages critical thinking, creativity, collaboration, and design thinking across multiple disciplines. An important role of STEM education is to help students develop skills that will empower them later on in the workplace. This includes helping students develop skills that foster:
    - ⇒ Critical thinking;
    - ⇒ Flexible thinking;
    - ⇒ Data-driven analytical inquiry;
    - ⇒ Design (interdisciplinary) thinking;
    - ⇒ Social responsibility;
    - ⇒ Productivity;
    - ⇒ Leadership;
    - ⇒ Teamwork;
    - ⇒ Collaboration;
    - ⇒ Communication. (Source: Techopedia)
- **Multimodality**. To understand multimodal learning, you first have to know the different modalities and their characteristics.
    - ⇒ Modes are channels of information. They include:
        - ▪ Speech

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

    - Audio
    - Written and print
    - Illustrations

  An example is that people learn from images by reacting to visual cues such as photos and graphs. People can also learn from kinesthetics by reacting to tactile cues such as actions and movement. Multimodal learning is teaching a concept using more than one mode (visual, auditory, reading, writing, and kinaesthetic methods). By engaging the mind in multiple learning styles at the same time, learners experience a diverse learning style that collectively suits all of them. Thus it is meant to improve the quality of teaching by matching content delivery with the best mode of learning from the student. (Source: eLearning Industry

- **Flipped Classroom.** It is a pedagogical approach in which the times and spaces inherent in the teaching and learning process are inverted: the exploration of content is first done before class by the students (e.g. through reading, video analysis, etc.) in a space that tends to be more individual than group-based; in class, students have the opportunity to interact with the teacher and with each other, in a fundamentally group space, in order to apply, develop, clarify the content previously explored. This inversion thus transforms the teaching-learning process into an interactive, dynamic, and personal logic. (Source: Bergmann & Sams,2014)

Other connected terms (pedagogical eclecticism):

- Adaptative teaching.
- Personal inquiry.
- Dynamic assessment.
- Crossover learning.
- Navigating knowledge.
- Learning through argumentation.
- Learning from animations.
- Learning to learn.
- Event-based learning.
- Learning for the future.
- Immersive learning.
- Open pedagogy.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

Indicative literature:

- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12. ACM Inroads, 2(1), 48–54. https://doi.org/10.1145/1929887.1929905
- Ben-Ari, M. (2001). Constructivism in computer science education. Journal of Computers in Mathematics and Science Teaching, 20(2), 45–73. https://doi.org/10.1145/273133.274308
- Bergmann, J. & Sams, A. (2014). *Flipped learning: Gateway to student engagement.* International Society for Technology in Education.
- Bialik, M., & Fadel, C. (2015). Skills for the twenty-first century: What should students learn?. Boston: Center for Curriculum Redesign
- Gómez, L. A., & Fischer, C. O. (2017). Android + Arduino usando o MIT AppInventor: Integrando Android, Arduino e o MIT App Inventor, para implementar a Internet das Coisas (IoT) (Portuguese Edition). Novas Edições Acadêmicas.
- Hill, D. R. & Brunvand, S. (2018). Gamification: Taking Your Teaching to the Next Level: A Guide for Gamifying your Classroom. In A. Ottenbreit-Leftwich & R. Kimmons, The K-12 Educational Technology Handbook. EdTech Books. Retrieved from https://edtechbooks.org/k12handbook/g
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. Computers in Human Behavior, 52, 200–210. https://doi.org/10.1016/j.chb.2015.05.047
- Romero, M., Davidson, A.-L., Cucinelli, G., Ouellet, H., & Arthur, K.. (2016). Learning to code: from procedural puzzle-based games to creative programming. Revista del Congrés Internacional de Docència Universitària i Innovació (CIDUI) (online journal), 3, 1–8. http://www.cidui.org/revistacidui/index.php/cidui/article/download/944/909
- Serrat, O. (2017). Storytelling. In: Knowledge Solutions. Springer, Singapore. https://doi.org/10.1007/978-981-10-0983-9_91
- Walter, D. (2014). Learning MIT App Inventor: A Hands-On Guide to Building Your Own Android Apps. Addison-Wesley Professional.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

- Wing, J. (2006). Computational thinking. Commun. ACM 49, 3 (March 2006), 33–35. https://doi.org/10.1145/1118178.1118215
- Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2017). Expanding computer science education in schools: Understanding teacher experiences and challenges. Computer Science Education, 26(4), 1–20. https://doi.org/10.1080/08993408.2016.1257418

Competences / Learning Goals

Key Competences

STEM / Personal, social and learning to learn, literacy, citizenship, responsability

Knowledge

Main concept: The role of low-code programming in public health issues

Programming concepts:

- Basic and Intermediate programming expressions**,** statements, procedures, and variables.

ICT concepts:

- Programming solutions development and application; low-code development environments; low-code development in public health *(topic: nutrition habits).*

Knowledge - outcome assessment:

1. Elaborates on concepts of computational thinking/science.
2. Recognizes and correctly outlines relevant concepts of low-code and block-based programming.
3. Easily outlines the importance of content creation-involved concepts.
4. Identifies and characterizes different programming languages and their applications.
5. Identifies and knows how to apply numerous basic/intermediate statements and expressions in MIT App inventor.
6. Justifies why low code is crucial to the future.

Skills (abilities/competences)
General:

- "Creactical skills" (Ohler, 2013) / 21st century key skills:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

- Communication: digital communication; digital literacy; traditional literacy; health literacy; public speaking; argue capability; learn to learn.
- Collaboration: working in groups towards a goal/to solve a problem/answer a question; collaboration.
- Critical thinking: perform reasoning and analysis to draw conclusions based on simple systems; decision-making process; problem-solving process; project-based thinking.
- Creativity: involves initiative, entrepreneurship, taking risks and learning from risks.

Specific:

- Developing, enhancing, and practicing computational thinking and technology-based projects.
- Finding, analyzing, and interpreting multimodal content to map basic/intermediate principles of low-code programming.
- Deeply expanding the 21st century competences.

Skills – outcome assessment:

1. Identifies and conceptualizes core and detailed skills that are required for programming.
2. Can demonstrate that multimodal, gamification and flipped classroom approaches are in fact, key for the future of education.
3. Sketches possible technological solutions for needs/problems of the healthcare market.
4. Can fully transform creative ideas into basic and intermediate programmable concepts using content creation tools.
5. Feels able to explain the benefits of using low-code development environments for real life problems connected with lifestyles.

Affective/Attitudes/ Behaviour (*beliefs)*

- Adopting a citizen developer role in society, as well as having social and personal responsibility.
- Pursuing the adoption of critical thinking and problem-solving attitude as an individual and in connection with the needs of the community.
- Engaging in more challenging programming challenges/courses to further develop his/her interest in STEM.
- Adopting attitudes that mitigate public health risks.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

Affective, Attitudes and behavior - outcome assessment:

1. Believes that low code is about innovation and creative problem-solving, that it should be explored in educational environments and that is a powerful tool for multimodal content creation.
2. Believes that technological skills are essential for effective citizenship and can lead to positive outcomes in educational, healthcare and business environments (e.g., fast development, scalability, simplicity, accessibility, low costs).
3. Believes that digital literacy translates into efficiency, access to things, knowledge, fulfillment, and happiness in personal and professional life, contributes to academic performance and improves student engagement.
4. Believes that health is the most important constituent of life.
5. Intends to further improve his/her digital literacy and programming knowledge to influence "STEM adoption" in his/her living environments and boost public health literacy.
6. Is committed to develop further his/her digital literacy in order to communicate easily on any subject and has a positive attitude towards it.

Learning goals and outcomes

- Uses low-code environments and content creation tools as creative platforms/extensions to express ideas and knowledge.
- Easily uses online tools to create multimodal content.
- Understands and characterizes the most important components of programming and uses them to create apps/games from scratch.
- Knows and communicates the main principles of good nutrition patterns, how much they may influence our daily lives, as well as their contribution for life expectancy and quality of life.

Assessment methods

- Outcome assessment
  - Quantitative – A questionnaire (in digital format)
    https://ec.europa.eu/eusurvey/runner/STEM-Tecnologia

    https://drive.google.com/file/d/1HuvJTCmGhPlf32A8hPPWdNueDBO73261/view?usp=sharing
  - Qualitative - students project: a. basic app/quizz building activity b. additional multimodal resources regarding public health and technological principles.
- Process assessment – *assessment of the teaching-learning sequence sequence* – observation grid: reaching the target audience, and extent; implementation of the scenario as planned; run

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

of the learning scenario as expected/organizational issues to be solved; duration of the teaching-learning sequence; number of people exposed; score for likeability – students ("how fun was it to do"/ how fun would be to do again/ how could it be better).

Content (relevant to learning goals & research topics)

STEM content

- Technical literacy.
- Basic programming expressions.
- Programming oriented math.
- Digital Literacy.
- Communicating science: Healthy lifestyles – nutrition habits.

Non-STEM content

- Teamwork
- Metacognition
- Proactivity
- Multimodality
- Autonomy
- Brainstorming/Mental Mapping.

Digital Learning Objects (DLOs) and Digital Educational Resources (DERs)

New (*developed by PAFSE team*):

Given the versatile nature of MIT APP Inventor and the imperative need to let teachers choose their own way of teaching their class, suited to student's needs, the PAFSE team, after discussion with pilot schools, opted by providing general animated, intuitive, and interactive *Noocs*[2] (*Nano Open Online Courses*), used in the Pilot workshop (all available in a e-me hive: https://files.e-me4all.eu/s/GpWkoyobw2DYdAp)

[2] NOOC are "nano" learning experiences that are specific, targeted to a certain skill and or competency, and can be disseminated in smaller, isolated ways.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

1. Animated, intuitive, and interactive *Noocs*[3] (*Nano Open Online Course*) using genially[4] concerning MIT App Inventor (Basic) and concepts to communicate science through technological content.
2. Informational multimodal resources (images, videos, sound, etc.) with content, challenges and solutions regarding basic low-code programming.
3. MIT AppInventor app(s) created by the team.
4. Questionnaires – quantitative and qualitative assessment of learnings - Student Interest and Choice in Science, Technology, Engineering and Mathematics (STEM Survey (adapted from Roller et al. 2018 and Faber et al, 2013); Student Interest and Choice in Technology; Informed Consent – Students; Informed Consent – Legal Representatives; Process Assessment – Observation grids, weekly meetings; Photos, Videos, Tasks; Final projects.

Available resources (link):

E-me hive with all resources: https://files.e-me4all.eu/s/GpWkoyobw2DYdAp

From other sources/high-quality platforms[5]:

Tutorials and examples to aid teachers prepare and train the students:

- MIT AppInventor:
  - Teach your students
  - Setting up your classroom for teaching App Inventor 2
  - Teaching an app inventor course
  - Hello Codi! (*app example*)
  - The MIT App Inventor library: documentation & support
  - MIT App Inventor - tutorials
  - MIT App Inventor - Beginner videos
  - MIT App Inventor - Nooc
- Block-Programming:
  - Block-based Programming in Computer Science Education
  - Block coding 101

[3] NOOC are "nano" learning experiences that are specific, targeted to a certain skill and or competency, and can be disseminated in smaller, isolated ways.

[4] Genially is the world-leader in interactive visual communication using low/no-code. It is an all-in-one online tool to create stunning presentations, interactive images, infographics, gamification, quizzes, breakouts, portfolios, etc. and enrich them with interactivity and animation effects in seconds.

[5] The majority will be included in the developed DER.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

- Nutrition habits[6]:
    - Referencial de educação para a saúde
    - Nutrition - CDC
    - Nutrition for Teens
    - Nutrition and teens
    - Take charge of your health: A guide for teenagers
    - Healthy Eating for a Healthy Weight
    - How dietary factors influence disease
    - Diet kills more people globally than tobacco and blood pressure

Observation:

- The PAFSE team provides examples of high-quality platforms that can be used by students to develop their app in the research project. However, teachers are encouraged to choose the resources they see fit, or even leave it up to the students, instigating their creativity and research capacity, since it is such a "mundane" topic. In case of the schools that have science/health teachers involved (in addition to/instead of ICT teachers), this is even more encouraged.

Teaching-learning activities

Principal target:

ICT classes /Biology classes /F.Q classes/Health Education classes (depending on the institution)

8h grade (+/- 13-14 years old students)

At least 4 classes of 40 min. (lesson 1- 4)

ICT teachers integrate other colleagues in the enactment of the scenario, as it aims to be interdisciplinary. The scenario provides the necessary tools for students to explore desirable behavior in an individual and public health perspective.

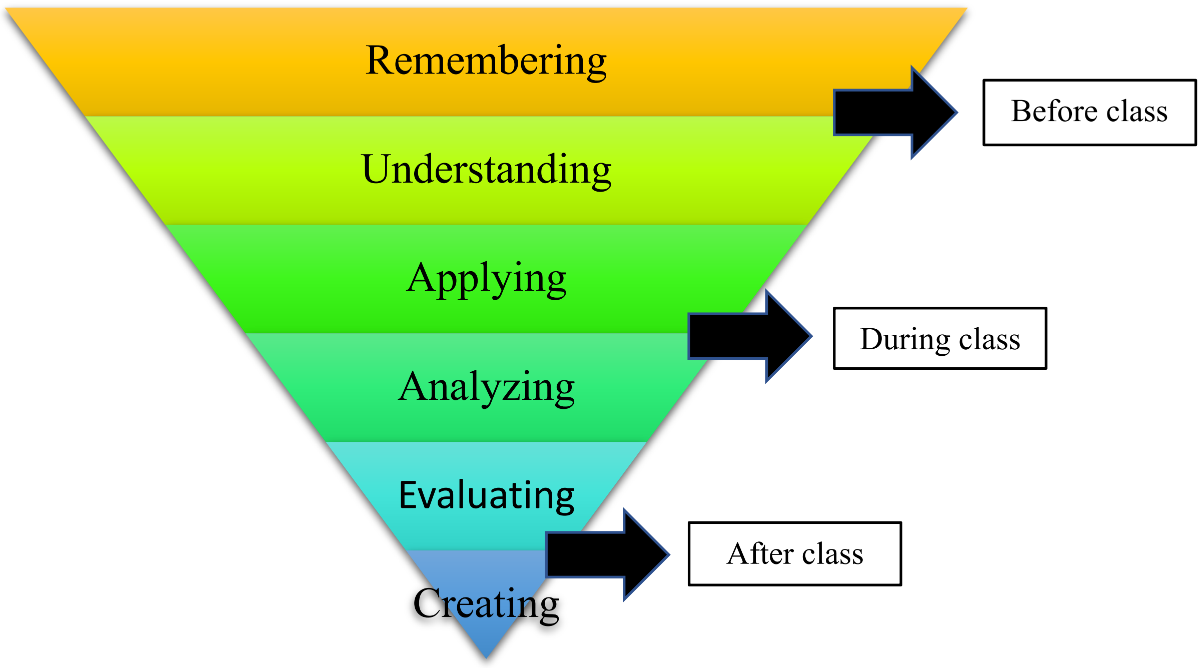
General note:

- This scenario focuses on learning by doing but also on learning by fully immersing on the topic. That being said, teachers are merely mediators and content curators, and support should be minimal. Students are supposed to solve the majority of their doubts by studying the issue and/or asking a peer.
- All the activities/theoretical aspects used in lessons will be available in the DER so you can all any information there.

[6] The topic is merely a suggestion, teachers are free to explore any health-related topic.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

- The lessons will have "Bloom's taxonomy" for flipped classroom as a background.
    - Students will have activities in which they will, independently and individually, explore specific (and always available) content at home and then bring findings/discussions to the class, creating a powerful sense of interaction and collaboration.
    - It is about building the knowledge themselves and then apply it at the classroom, in group.



Bloom's Taxonomy for "Flipped" Classrooms; version revised by Lorin Anderson

<u>Lesson 1: Introduction to key concepts</u>

<u>Learning objective</u>: In the starting session, students will be exposed to various theoretical concepts (discussed in the workshop) they will need to use MIT App Inventor successfully and attain the scenario goals. The activities performed during this first lesson aim to engage students in these topics and explore some preliminary ideas.

Topics to be explored: Content Creation; Storytelling; Gamification; Programming; Computational science.

⇒ The teaching-learning script starts with a video play: Programming as a kid

- The teacher will project the short video to the classroom.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

- When it finishes, pose the question: "What about you? Is that how you feel about programming?" and ask several random students to answer (you can also ask them to raise their hand).
  - If students responded positively to the question, ask: "What about gaming? Do you like gaming? Let's take a look at other related short videos"
  - If students responded negatively to the question simply generate some discussion with different points of view and tell them that after the scenario enactment, their view on programming will be a lot different and then ask "What about gaming? Does that seem more interesting to you? Let's take a look at other related short videos"
- Show them 2 more videos:
  - Gamification - definition
  - Learning through gamification
- At this point they will be very excited with this topic so profound it now.
  - Correlate gamification with storytelling and show them this short video.
  - Present them the detailed steps to create a storyboard
  - Correlate with the topic of "Multimodal content creation" by providing real examples.

⇒ Now start developing on the importance of digital literacy, pointing the fact that technology is everywhere, and we need to integrate it in education and business matters, as well as discussing the topic "programming languages" and "block-programming", analyzing what they already know, as well as giving them some "basics".

⇒ What is MIT App inventor and what will they use it for.

- Presentation
  - The best way to understand App Inventor is to use it.
  - The implementation of a program is done in two parts:
    - The creation of the user interface and the choice of resources that will be used in the application,
    - then block programming that uses the components (events, properties, procedures) associated with the components defined in the first part.
- SETUP
  - Activity: Each student should successfully SETUP independently.
  - Let them explore the platform until the rest of the class and provide them with some guides for consultation.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

    - Explain on what they will work on and attribute and elaborate on the topic "nutrition habits".

⇒ Assign obligatory homework: Discovering code

    - Since this is a homework activity, you won't be able to guide the students, however, that is also the goal because it is where flipped classroom jumps in. Students will have the opportunity to acquire knowledge autonomously and without direct exposition of theory and will, afterwards, clarify doubts and discuss about it in the classroom.
    - Ask students to open this website, watch the video and complete the tasks (phases of a game).
    - In less than an hour, they will become familiar with block sequences, conditional connections and loops. This will get them started on practicing basic notions of block-based programming and they will like doing this activity.
        - Tell them to setup the page to PT-PT since it's available, both video and game.
        - In case there are some fast and very curious minds or even has a taste for flappy bird, you can give them another example.
        - Ask them to write down (in their notebooks) any doubts, comments and general observations regarding the learning experience. Some questions to ponder on:
            - What did you learn about how apps and games work today?
            - How do you feel knowing you programmed, you gamified.
    - Challenge students to share some aspect of their Hour of Code experience on social media using #hourofcode as a way to lend their voices to this worldwide movement. Students can share their game, images, videos or just their thoughts.

⇒ Questionnaires – quantitative and qualitative assessment of learnings - Student Interest and Choice in Science, Technology, Engineering and Mathematics (STEM Survey (adapted from Roller et al. 2018 and Faber et al, 2013); Student Interest and Choice in Technology

Lesson 2: Discovering coding 2.0

Learning objective: In this session, students will keep discovering code, this time, with the help of the teacher and a lot of reflective work will be simultaneously carried out. The plan takes up that of the courses of Professor Ralph Morelli and David Wolber. The pedagogical logic is that called *"BCCC" for "Build, Conceptualize, Customize, Create*": we start by doing (copying), then

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

conceptualize (by experimentation and error research), customize existing functions, and finally create new applications, guided and then in free flight.

Topics to be explored: Block-programming principles.

⇒ Homework "correction"

- Start by talking a little bit about the work they did at home and what type of skills were being used in the process.
- Continue by elaborating on the expected learning outcomes topics
  - Define "coding" and "computer science"
  - Identify key computer science vocabulary
  - Make connections between computer science concepts and the real world
  - Identify places to go to continue learning computer science and coding
- Division of the class in small groups (4-5 people or as the teacher sees fit) in order to stimulate collaboration and capacity of exposing and discussing ideas, which will lead to a moment reflection on what they wrote about it.
  - Ask students to share their game, app or final product with the peers, pass by to take a look.
  - Give them a few minutes to discuss and then ask each group to present some bullet-points about the experience.
  - Listen to their feedback about it: clear their doubts; and listen to what kind of observations they have/how was the experience for them.
  - Ask how many felt the experience was easy/hard; pleasant/unpleasant.
- After discussing
  - Celebrate! Give them some certificates to keep it fun and make them proud of their work.
  - Tell them they can use the same website to practice in case they got interested and want to explore more.

- Maintain the groups formed and ask them to quickly scan MIT App Inventor Interface.
  - Give them a couple of minutes.
  - At this point they will be confused, as expected, so next provide them with some links Built-in blocks; Component reference and
  - Ask them to start creating the app "Paintpot" following a tutorial (if they are feeling particularly adventurous they can also do part 2); https://appinventor.mit.edu/explore/ai2/paintpot-part1; https://onvaessayer.org/appinventor/baseApps/paint.php
    - This tutorial will lead the students to:
      - define an application by how it reacts to a list of events (event programming or event driven programming),

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

- choose the components you need and layout them (Design)
- buttons to choose colors, line thickness, erase or take a photo,
- a frame (canvas) to draw and display the photo,
- horizontal arrangements to layout,
- a camera (camera).
- describe what the application should do for each event and then code this behavior in a script associated with that event,
- define and use variables, for example for the size of points.

▪ The steps in the realization of an application:

1. Initial analysis: define what the application will do, for whom, what need it meets.

   In this phase, you need a pencil, a paper but also friends to whom you can submit your ideas. It's not programming, but it's important.
2. The development:
   1. Design of the user interface and the choice of resources,
   2. Programming: scripts or sequences of instructions (or blocks) that follow an event,

It is most often an iterative process, ideas improve as they are realized, but be <u>careful to keep your priorities in mind</u>.

▪ Check VERY often what it looks like on your phone, test every step, don't wait until you finish a big piece,
▪ A mistake or a small accident quickly happened... especially when you are in the most hurry. Regularly save intermediate steps or versions, especially with checkpoints.
▪ Transfer your program often to mobile, preferably work in connected mode (or interactive debug),
▪ Test to develop and check proper functioning.
▪ In the design part:
  - Choose/find the components
  - Does the choice of component names have a theoretical or practical impact on the next programming phase? It has a practical impact because the name will allow us to remember what each component corresponds to when you write the program. For example: "ScanButton" allows you to remember that this is indeed the scan button. However, it doesn't matter to the program. We could very well have called this button "Michel" or

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

"MaBicyclette". You can rename it and turn that the program works exactly the same. The difference is practical, neither computer nor theoretical.

- The design allows you to choose the main properties of the components. It will be possible to modify these properties later in the program, but it is not always necessary.

▪ In the Programming section: the events and blocks used:

- What events have we taken into account?
    - ▪ user intervention: Clicks on the buttons, frame touched or dragged,
    - ▪ and the events triggered after smartphone functions: the arrival of the photo after shooting.
- Blocks have several shapes: which ones and what do they correspond to?
- the same goes for the color?

1. Permanent transfer of the program to the phone or tablet: You have used two modes of communication with the mobile phone:
    - the first (connected mode), in the development phase, where the application remains controlled from the PC, which allows quick testing and modification, the second (build or build mode) where the application is downloaded to the laptop and works independently.

This app development isn't going to be easy since there were more basic apps to construct but you are there to provide as much help as they need.
If they don't finish in time, tell them to finish at home without worries.

Lesson 3: Practice makes perfect

Learning objective: during this lesson and as wrap-up moment on the "app creation" world, students are invited to challenge themselves and take it to the next level. Basic concepts where explored. This is the moment to explore even more.

Topics to be explored: Intermediate block-programming principles.

⇒ Activity 1: *Discovery Time -- App Experimentation*

- o Play around with the following source code for the app Dancing Llama by experimenting and completing the following tasks:
    - ▪ the moveLlama procedure block. See how the corresponding blocks with that procedure also change names.
    - ▪ Rename the numberOfDances global variable block. See how the corresponding blocks with that procedure also change names.
    - ▪ Answer the question: Why does the numberOfDances displayed on the label only increase when the timer goes off and not when the phone is shaken?

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

    - What happens to the label when you shake the phone 3 times in a row?

- Randomness is an important and common task in computer programming. To decide whether or not the ghost in Pacman should turn left or right at an intersection is determined randomly or whether or not a mushroom should appear in a game of Mario Kart. The generation of random numbers can be used to make these decisions. If you would like the Pacman to turn left
- approximately 25% of the time and to turn right approximately 75% of the time, you can use blocks like this:
- In this "Pacman" program, there are two procedures: turnLeft and turnRight. A random fraction [*between 0 and 1*] is used to determine which one should be called. If this fraction is less than or equal to 0.25,
- Pacman will turnLeft. Otherwise if it is greater than .25, Pacman will turnRight. Although it is random what will happen, because there is a greater possibility of numbers to randomly choose that lie in the range [.25-1.0], it is more likely that Pacman will turnRight in this program.
- Basic (and advanced) algorithms are used all the time in Computer Science. Basic ones are used in app building to solve problems. In game apps, there is an algorithm that tells the score to increase by one when the mole is hit. Essentially an algorithm is a set of rules or instructions that defines a sequence of operations.

⇒ Activity 2: Divide the class into small groups and divide these 3 apps between then, assign randomly.

- In the end, a group of each app will present it, succinctly explain the process and give the templates to the colleagues (so they have a lot of material for the research project).
- Particularly, their assignment is to play with the source code and corresponding app for the following three games: Mole Mash, Get the Gold, and Space Invaders. Take notes about they like/dislike about them. Have them look at the blocks, components, and the design of the app.
    - Mole Mash
    - Get the Gold
    - Space Invaders
- Make them think about their observations of these games as well as develop their own game app. A game is defined as structured play. Sometimes games can be work as in the case of professional athletes. But mostly games are for fun. The key components of all games are goals (to hit the mole), rules (you only get points if you hit the mole), challenge (the mole moves randomly across the screen), and interaction

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

(the user plays the game on the phone by touching the screen with a finger). This game can be a modification of any of the above apps or a new creation of your own.

⇒ **Simpler Alternative to all activities (tested in the Pilot)**: Let's keep them learning by doing – Creating the "survey" app, available in "Sessão nº 2_12dejaneiro":https://files.e-me4all.eu/s/GpWkoyobw2DYdAp

Lesson 4 and further: Practice makes perfect

Learning objective: during this lesson and as wrap-up moment on the "app creation" world, students are invited to challenge themselves and take it to the next level. Basic concepts where explored. This is the moment to explore.

Topics to be explored: Intermediate block-programming principles.

The last assignment was a big one, so they will probably need a lot of time.

This lesson is free for practice and continuing developing games.

Lesson 5-forward: The final countdown

This is the **School Project** described below.

Learning objective: In these final sessions, a recap regarding the major practiced concepts concerning block programming is made, as well as of the used methodologies. In addition, students are prepared to the student's research project.

Topics to be explored: Mental mapping, Pitch skills

⇒ Present the concept of a mental map by showing one representing the lessons.

⇒ Activity 1: Ask students to elaborate one, on paper, describing what they learnt.

⇒ Aiming to prepare the students for their final project, a presentation called "O meu Pitch em 5 p's" (My pitch in 5 p's) will be given, teaching students how to publicly present projects.

- How to communicate science/health using these 5 pillars:
  - "Priorizar" (Prioritize).
  - "Pesquisar" (Research).
  - "Planificar" (Plan)
  - Personalizar (Personalize)
  - Produzir (Produce)
- Provision of storyboard templates to prepare the pitch.
- Explanation of what will be evaluated.

These are moments of creative freedom, where students apply the programming concepts learned during previous sessions in a project linked with hygiene habits. There is freedom to work with

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

the MIT AppInventor platform or use paper modeling, with the aim of presenting the project at the end of the sessions, or in another event created for that purpose.
The projects will cover all the tools and themes explored. Students may choose, within the topic of hygiene habits, which sub-topics they want to explore and focus on. Regarding programming skills, they should follow a simple framework to create their own apps, so that they can be as independent as possible.

Supplementary learning resources and educational activities

1. Regarding the school Research project:
    - Production of multimodal content – Students can transfer and use the knowledge acquired to other classes.
    - Public health reflection.
    - Competition – reward of the best app ideas.

School Research Project (Ciência Viva)

Topics

Importance of digital literacy and real-life implications.
Basic/Intermediate technical features and principles of programming solutions development.
Possible applications of mobile applications in public health (e.g., in the promotion of nutrition habits).

Research management, design and administration

Challenge: Content exposition creation on one topic involving "nutrition habits", promoting not only technology, but also public health.

Method *(summary)*: Lessons 5 to 7 will be dedicated to the school research project. Students are, as usual, organized in groups and each group addresses the practiced programming and technological concepts and connects them to health.

Development process:

The project is based on the use of technology to create scientific artefacts. The five-six sessions will be lightly supervised by the teachers and developed by the students, with scheduled moments for checking the work development.

Groups of students will be instructed to create an interactive game or tool app that explores the topic of hygiene habits in some way, as well as some other resources (of their choice) that they see fit on the same topic:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

During session 1, students are presented, not only, with software to use for content creation but also the norms to follow:

1. Each group should have, at least:
    a. 1 app on the theme of healthy habits (they can use existing templates or build from scratch and there is no need to be similar to those they tried, it's full on creativity)
    b. One other multimodal resource on the theme of healthy habits.
        i. It can be a mental/conceptual map, a quizz, a presentation, an interactive resource, an infographic, a story, a video, etc.
    c. A short portfolio with all the created resources.
2. Each group is required to:
    a. brainstorm a project idea, develop drawing(s) of the app/game on paper, creating a storyboard.
    b. meet regularly with the teacher to discuss the feasibility of the app and if necessary make any changes to their plan.
    c. Present a short (till 10 minutes) elevator pitch of your project idea to the class.
3. Create a portfolio (free structure, let the student be creative) write up of your project.

Finishing session 1, the groups are created, and ideas start being discussed.

During session 2, the teacher will pass by the groups to collect ideas and topics and, if valid, students can start working on their project. The teacher will provide all the needed help, even if that means that he is an contributor to the project.

From session 3-forward, the students will actively work on their project and are encouraged to exchange ideas with other groups.

Teaching-learning process milestones:

1. Students will be able to propose basic/intermediate programming solutions.
2. Students will be able to communicate the findings, motivations and limitations of various solutions considered in the work process.
3. Students will be able to identify and communicate the importance of digital literacy/end-user development in public health and citizenship.

Teaching-learning process for school project (summary):

1. Development of multimodal materials.
2. Mobile Applications development.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006468.

3. Presentation of all the resources created in the open schooling event, where students will be advocating better conditions for their community and show their relationship with public health and low-code environments.

Organization of the open schooling event:

1. Each project output (portfolio) is presented by the students in a community setting (e.g., exposition center, municipality, science fair) with appropriate/pre-prepared environment (computer and smartphone with the MIT App Inventor installed).
2. Students do a pitch on how mobile solutions can be used to address public health, like the case of good nutrition habits. Technical speeches to motivate peers to new technologies and technological environments. Students will also be advocating better conditions for their community and show their relationship with citizens health.
3. Students, parents, school community and relevant local stakeholders attend the event and recognize that mobile solutions can be used to address real life challenges, public health ones, and others. They also get high-level understanding on strategies to minimize the phenomena and how they may have an influence on the relevant settings (e.g., home, school, workplace, community).

Data Analysis and Reporting

Content Analysis.

Multimodal resources.

Portfolio development.

Target Audience for Recommendations

School community and local stakeholders: students, parents, municipalities, engineers, public health authorities, and local enterprises.

Public Debate and Recommendations (based on research results)

Presentation of the resources produced by students in a community setting and dissemination of evidence recommendations via social, community and conventional media.

Discussion and feedback.

Attribution of the prize of “best apps”.

Main partner responsible: UM (University of Minho)

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006468.