

With a background that is often in the area of education and not programming, many students resist understanding the meaning of programming in this profession since it is not essentially programming.

Understanding complex concepts:

Many students appear to understand better with visual representations and interactive simulations to understand the abstract principles of programming more effectively.

Engaging Learning Experience:

Students of any course, especially this one, tend to respond better. The use of gamification elements in instructional design makes learning programming smoother, using, for example, interactive activities and laboratories where they can create and actively participate in the learning process

Real-world application of skills:

Understanding how this knowledge will be applied makes all the difference in engagement. Incorporating case studies and real-world projects allows students to apply programming concepts to practical scenarios.

This approach helps students understand the relevance of the content and prepares them for real programming tasks in professional environments.

Feedback and follow-up:

Providing instant feedback on coding exercises allows students to identify and correct errors in real time. They can monitor their progress promoting a sense of accomplishment and continuous improvement.

Preparation for industry demands:

Reflecting current industry practices and trends, students with the relevant skills sought by employers are more likely to be hired.

When considering these benefits from this perspective, students of educational technologies understand that they can create programming content that is not only educational, but also engaging, adaptable and aligned with the diverse needs of students.