Computer architecture simulation on the server–side for online evaluation purposes

Javier Romero[1], Felipe Romero[1], Gerardo Bandera[1], Luis F. Romero[1],

Resumen— **Due to the global COVID19 pandemic, in the last few months there has been a dramatic change in the educational context where lecturers around the world has forced to solve academic problems that immediately before this crisis were completely unthinkable. The online assessment has been the main issue that has generated the most stressful situations (and not only for students, but also for teachers). In this work, we present an exam prototype for Computer Fundamentals modules in high educational levels. This tool essentially consists of variations and combinations of questions mounted on a Web server, and which, being supported by a set of simulators that implements hardware simulators, is able to generate a huge number of different examination proposals on the server–side. On one hand, this prototype simplifies the task of monitoring by the teacher, since the possibility for two students to have similar exams is null, in practise; on the other hand, it significantly reduces the student's stress, since this tool let them to have a countless number of exam samples previously, that they can practice with. In the 2019/20 academic year, where this tool has been used for the first time, has derived in a clear increase in the percentage of students who have passed the course and a very high degree of satisfaction of students and teachers. They are some of the indicators that remark the advantages of this prototype.**

Palabras clave— **Computer fundamentals, online evaluation, hardware simulator, COVID–19 confinement.**

I. Introduction

THE evaluation within the University education process is that "necessary evil" that many students would like to suppress. But it is completely essential, not only because of the overcrowding in classrooms since, in a certain way, we must value the work done and the level of knowledge acquired in an individual way, but also because through it, the student is bound and motivated to seek a greater degree of knowledge, while learning to self-control in difficult situations. Establishing the way to carry out the evaluation requires special attention, and more than to determine the degree of knowledge of a student, is to estimate his ability to use them. Moreover, in most technical studies, the contents of the evaluation tests should be prepared avoiding questions of memory nature. They should include aspects that serve to detect the ability to apply, generalize or synthesize concepts.

Another important consideration affecting assessment is the fact that too many students study solely for the purpose of passing the course, and not for learning. To compensate this typical trend, we must first pay special attention to the form of assessment, so that both objectives (to pass the exam and learn) can be achieved simultaneously. Secondly, it is absolutely necessary to establish surveillance mechanisms during control tests to persuade students that they intend to pass without learning, to put it in an euphemism. It is not necessary to remember that the student who cheats, by favoring mediocrity, not only harms himself, but also those who pass the exams on their own merits (i.e, the "pass" is devalued) and also the entire educational system. When classroom overcrowding is combined with the need for control, the joint examination in a supervised classroom has been, until now, the main feasible mechanism of a fair evaluation, from the point of view of a controlled situation. However, when the presence of the student in the classroom or the examination room is unfeasible, as has happened recently during the COVID-19 virus crisis, lecturers have found themselves in a situation in which justice, which should prevail in the evaluation process, is completely overcome by circumstances in which the roguery, and the consequent penalty to the good student, have a good chance possibilities of succeeding.

To avoid this unfair situation, evaluators have many tools that current software technology provides, such as virtual rooms via videoconferences, tests including random questions, results calculated from the Moodle tool [1], and even the numerous anti-plagiarism tools, whether generic or ad-hoc. However, the main ally of justice in the evaluation process in a confinement epoch is, undoubtedly, the time: when a student has a very tight time frame to take his test, is more difficult to contact or receive help/tips from other colleagues to perform his own exam, as he barely has enough time to solve it for his own.

A. *Proposed evaluation system within the contex of the COVID-19 pandemic*

During the recent period of confinement, which in Spain began —with academic effects— on March 13, 2020, has continued until the examination period in June, and now we are in it again from November 2020 to summer 2021, lecturers of a Computer Fundamentals module of the School of Industrial Engineers of the University of Málaga (Spain), jointly with some current and former students of the module, have developed a novel assessment system. It is based on a set of simulators [2] that combines randomness in the questions and the time available to take the exam, which makes it practically impossible that a student can get help/tips from other colleagues who are taking this test simultaneously.

[1]Dpto. Arquitectura de Computadores, Universidad de Málaga, e-mail: `jr@uma.es, ipe@uma.es, gbandera@uma.es, felipe@uma.es`

The simulators are completely transparent to the student, and are used, firstly, for the generation of a unique variant of the exam, secondly, so that these generated variants include congruent and balanced questions, and thirdly, to obtain an automated correction from the system. We would like to remark that during the examination, student only receives a single exam version. In short, simulator acts to generate exam versions and to correct them automatically, but in any case they do not intervene or access during the exam time.

On the other hand, each version of the test is nearly random, and the number of combinations on every partial test carried out is so high (it far exceed 10^{48} combinations). This diversity of proposals generated from the system made us to take the decision of using this tool in an alternative way. It will be not only used to make student examinations but also we let them to do practices before the examination day. Hence, students are able to try over and over again, avoiding the painful "surprise" of not knowing which will be the exam structure. With this, an increasing of the degree of training is expected.

B. State of the art

Distance Education is not new. Its origin dates back to the last part of the 19th century and the beginning of the 20th century, when people could not travel long distances to Higher Education institutions. In the last 20-25 years, and thanks to the rise of the Internet, many institutions have been promoting their distance education offers. This has meant, on the one hand, the possibility of having new channels and/or potential students and, on the other, the opportunity to transform teaching to confront with guarantees the highly competitive panorama in which we are [3]. In 2000, Volery and Lord identified in [4] three critical aspects when it comes to achieving quality distance learning: i) used technology (ease of access and navigation, friendly user interface), ii) the type of instructor (attitude with the students, technical ability with tools, class interaction, etc.) and iii) the technological skills of students.

It is important to highlight that, with online education, the role of the students goes from being a mere passive recipient of knowledge to being an active agent of learning [5]. In this sense, Draves presented in [6] ten reasons why distance learning is more popular and better, cognitively speaking, than face-to-face learning. Regarding the teaching staff, Medina and Miranda [7] determined the characteristics that make a teacher more suitable to achieve a great acceptance of the students in courses that go from being face-to-face to synchronous online. The results show that the best teachers are usually young, with excellent technological skills, ease of interpersonal interaction and good social skills. Experienced teachers with good technological skills and excellent teaching techniques also stand out.

In 2000, Hmieleski and Champagne demonstrate than about 98% of course evaluations were paper based (see [8]). From then, online courses evaluation has become bended in many universities while much research has been done to detect advantages, disadvantages, similarities and differences. An interesting study between research works comparing online and face-to-face evaluation methods is presented by Morrison in [9]. He remarks that caution has to be exercised in assuming that online and paper evaluations will necessarily yield similar results, even when the same instrument is used with the same population. Whilst this does not indicate whether one version is preferable to another, it suggests that reactivity to the medium might influence the results and, hence, the weight that can be placed on them. Ultimately, decisions on whether to opt for online or paper evaluations might be taken on grounds of a range of cost savings rather than for educational reasons, and both the literature review and the data in the present study indicate that, when time and timeliness are at a premium, these are important considerations.

There are a lot of comparisons of assessment results in literature using online and face-to-face exams. Many of them present similar results for both methods [10], [11], [12], while there are others that show just the opposite, that some types of exams produce better results than others [13], [14], [15]. Stowell et at show, in [16], that online assessments generate a significantly lower number of students presented than the face-to-face case (about 20%). This may be due to the anxiety that this type of evaluation produces in them, and the knowledge that teachers lose control of the conditions in which each student takes the exam: they can do it without ever having been to class, with the help of another partner, in group, etc. All of this can make them think that this type of evaluation will reduce their rating.

Recently, and due to the COVID–19 pandemic that we are suffering worldwide, some authors analyze the massive and widespread use of online teaching and assessment, as well as the results and consequences that this is producing. The importance of having a good distance education system available would imply being able to solve times of confinement like the ones we currently live in. This will allow a large number of students from all over the world, including those from developing countries or even the third world, to enjoy the right to education with many guarantees. In this sense, Basilaia and Kvavadze present in [17] how the use of a distance education tool affected in a European country. On the other hand, [18] analyzes the psychosocial disaster that a time of confinement can produce in the population of a third world country if teaching is degraded. Finally, George in [19] demonstrates that the use of appropriate online strategies for teaching and assessment during COVID–19 prevents students' performance from deteriorating. At the same time, it analyzes the main benefits of this type of methodologies and shows some examples of possible online exams.

II. Randomization in questions and simulation

The use of randomness questions is as old as the tests themselves. Either one way or another, teachers have resorted to a random change of some element in every exam question in order to achieve necessary changes in tests to get the student to study the concept instead of the answer. In most cases, the random (and variable) part of the question is obtained from a discrete set of elements, which can be numbers (e.g, "How much is $4 + \{7, 8, 9\}$?"), boolean values (e.g,"The 16's complement of 5 {is, is not} 11"), or even, colors (e.g, "If the traffic light is {green, red, yellow}, can I cross the road?"). Normally, for each element of the set (which we could call "input") there is a different answer that can be either calculated by means of a mathematical expression, or in most cases, it must be established more or less manually. It is clear that this second case is the one that requires special attention, as it surely represents the overwhelming majority of examination questions.

Except for those cases in which the maximum size of the input set (variable elements in the question) is very small, it is normal that lecturers, specially in the field of Computer Sciences and Engineering, need to calculate the corresponding result (i.e, output set/responses) of each question using a more or less considerable time. And it is precisely for this reason that the input set is usually small (and usually comes from a collection of previous exams). But what if we can automate the calculation of the answer?. Suppose a machine/computer where exams are automatically designed using tools (simulators) that allow you to calculate the output torque in a complex system of gears and motors. Obviously, the teacher could produce his own tables rather quickly, in order to have as many input layouts as there are students in his course. Even in this case, the teacher's effort and time invested are high enough not to resist the temptation to reduce the number of cases. But what if we go further? What if the editor that generates the exam text incorporates a specific simulator on the content of each exam question? In this case, the set of input elements could grow by several orders of magnitude without the teacher needing to determine the specific answer to each input set, since the simulator can calculate non-trivial solutions immediately.

A. The CASIUM simulators

In the specific case of the Computer Fundamentals course, which has served as an improvised training ground for this work, the set of contents was defined according to the recommendations of the ACM [20]. Hence, we incorporates in our tools a highly varied course subjects: the binary representation of information, instruction set architecture, computer data path design, memory information storage, memory hierarchy, input and output systems, and even some operating system issues, such as managing CPU usage or memory virtualization. Fortunately, in the months prior to confinement, a group of teachers and students of the course began to develop a set of specific simulators for each of the aforementioned subjects, supported by the *CASIUM* project (Computer Architecture Simulators – University of Málaga) [2]. In a first step, the purpose of these tools were not to evaluate the student, but to facilitate learning. Thus, simulators were developed for each of the main parts that constitute the architectural foundations of a computer. They have been classified into 5 modules, and some of them are shown below:

Module 1: Data Representation:

a) Alphanumeric encoder: This tool has been designed to study the encoding and decoding of characters of alphabets from around the world. It has been designed so that the student understands the difference between the different ways of encoding information (ASCII, ANSI, ISO, UNICODE, etc.), as well as the problems derived from an incorrect texts decoding.
b) Numeric encoder-1: This module also include 2 tools for non-real number representation (natural and integer): Two novel encoders, with a detailed display of the encoding process for the most used non-real number computer representation formats (BCD, natural binary, two's complement, etc.).
c) Numeric encoder-2: The last tool of this module is related to real numbers. It shows the floating point number encoding and decoding process using the IEEE-754 format, the most popular real number internal representation in current microprocessors.

Module 2: Basic Digital Electronics:

a) Multiplexer: The first tool of this module simulates the most used combinational component in the microprocessor design. Its operation is very simple and we only implement two parts: a signal decoder and a bus decoder.
b) Registers and Counters: This simulator is able to teach students about the operation of a simple register, where the binary information is stored, how counters work, and why timing is required in sequential circuits.
c) Adder–Subtractor circuit: This tool was designed to work using three different modes jointly with several combinations. The work modes are: adder mode (basic), subtractor mode, and adder–subtractor mode. It also incorporates a viewer of the inner full-adder circuits (Fig. 1).

Module 3: CPU Components:

a) Instructions Memory and Control Unit: The first tool simulates the first step of every instruction execution on any processor. It shows how the program counter (PC) output bus is used to select (read) an instruction from memory (small and randomly started). The selected instruction

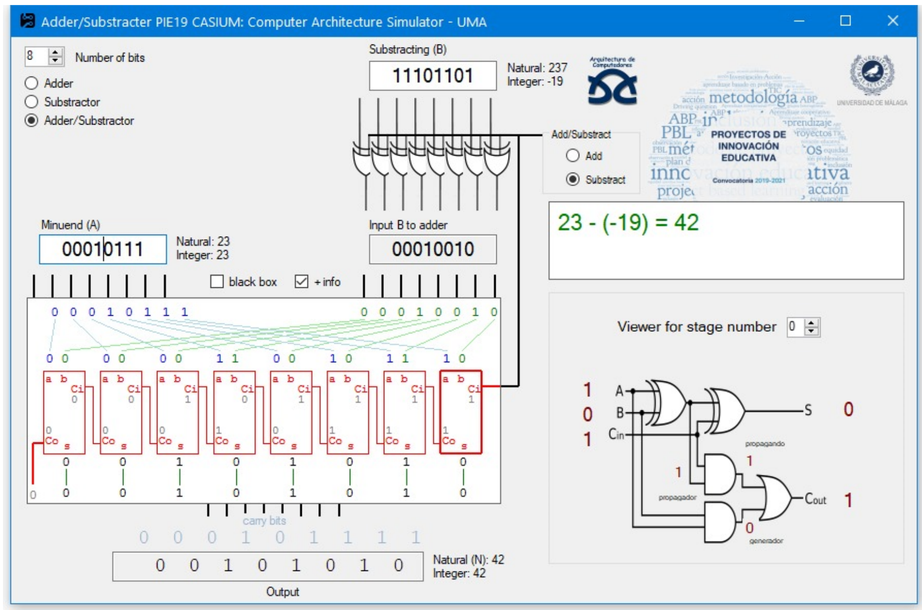

Fig. 1. Main screen of one of the eighteen simulators developed for the subject (adder-subtractor module, which incorporates the simulator of a full adder).

will appear on the circuit's output bus. It is well known (in the knowledge area, obviously) that the instruction is nothing more than a binary sequence, indistinguishable from any other set of bits. Therefore, the sequences must be interpretable as instructions and not as data. For this reason, in our tool, the memory will be two working modes:

- In basic mode, memory only stores binary sequences. In this case, nothing would differentiate it from a data memory, except the implicit property that it is "read only".
- In decoded mode, the binary sequence is interpreted as a MIPS or ARM instruction (depending on the configurable option).

b) Registry Bank: Undoubtedly, this tool could be the most important one. Taking into account that the registry bank participates in two of the five stages of a MIPS processor (an essential pillar in the teaching of the Computer Architecture area), a perfect understanding of its operation is essential for a good learning of the course. A bank is simulated with 32 registers, of 32 bits each (randomly initialized), with two read ports and one write port.

c) Data Memory: The programming of this tool reuses many of the classes and methods that are used in the corresponding ones for the instruction memory and the register bank. It has two modes of operation:

- ROM mode, with read-only accesses.
- R/W mode, where read and write operations can be used.

Module 4: Global Simulators:

a) MIPS microprocessor: This tool seems to be only a combination of many of the previous explained tools in fact (Arithmetic Unit, Instruction Memory, Register Bank and Data Memory). However, the complexity of this module goes much further, since, in this case, the inputs and outputs of the four previous modules are linked. In fact, the set of tools constitute a complete simulator of the reduced MIPS processor.

b) Delays in the CPU: The goal of this tool is to demonstrate how signal propagation delays on a digital circuit influence the clock frequency of a given microprocessor. It is proposed to use as a reference a CPU based on a simplified version of the MIPS processor, that is widely used in the teaching of large set of courses about Computer Fundamentals and Architecture.

Module 5: Input/Output and Operating Systems:

a) Preemptive multitasking and time quanta: A microscopic view of how the operating system deals with multitasking using time slicing. Any process can be blocked due to interrupts (Fig. 2).

b) Interruptions and Daisy Chain: Three devices can sen their external interrupts, which may be masked individually or globally. The acknowledge response to interruptions is managed by a daisy–chain module.

c) Wator: A population dynamics simulation of a toroidal ocean, using multi–threading and high intensive CPU usage, which is used in combination with the process explorer of the system to teach preemption and multitasking.

Many of these applications (compiled for Windows systems) have been published on the Microsoft Store platform for free, to easy the distribution among students [21].

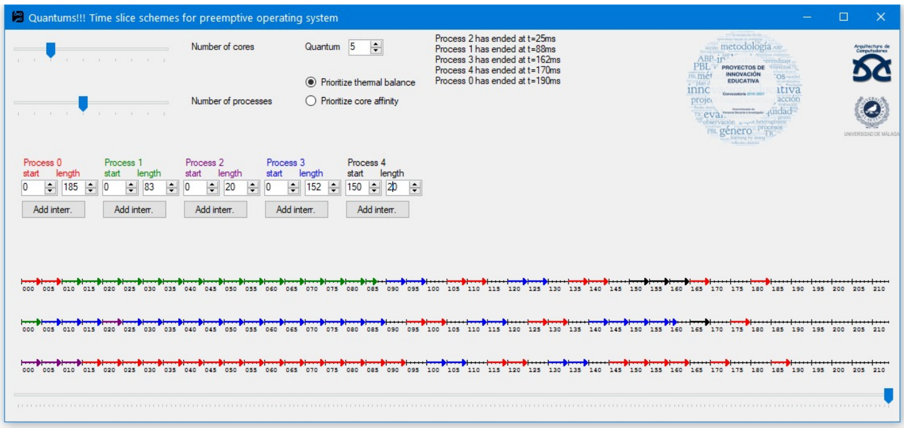

Fig. 2. Quantum! A simulator for the distribution of CPU time slots between different processes using preemption, in a multi-core architecture.

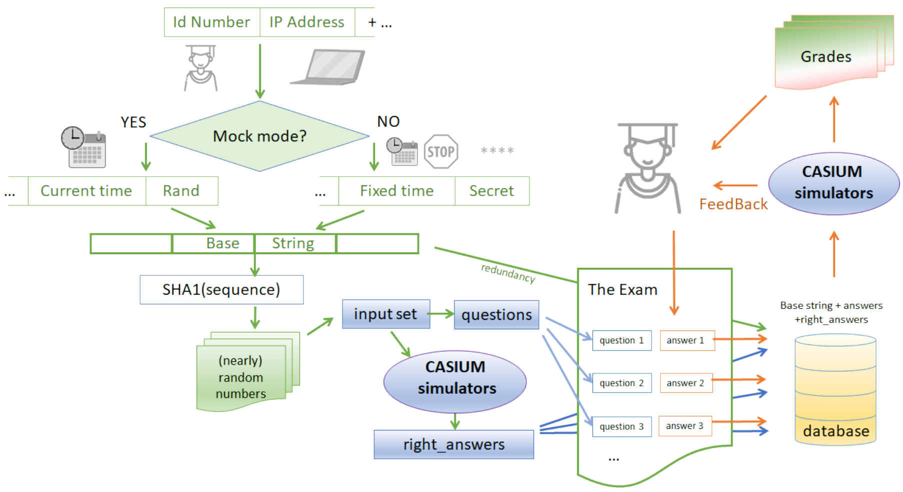

Fig. 3. A simplified scheme of the CASIUM test system

ID number	Date/Time string	Public IP	Private IP	Secret
25094239H	20210923113345	083177045033	192168001031	*******
sha1('25094239H20210923113345083177045033192168001031*******') = 0100110100011011011011111111110110100111001000110011100000111101011001010010111010010100110011010111000001110110000010111110010111 00...				

TABLA I

(Nearly) random values calculation

III. Test design

The design of every test with our tool is quite simple (a scheme is shown in Fig. 3). Each exam statement, —for a given student, in a certain place (IP address), and date/time— is retrieved from an URL after entering his ID card number. The resulting page, which is build using PHP and Javascript languages, generates an exclusive test from a set of base questions. The generated exam can operate in two main modes: First, in a *training mode*, where each page reload produces a different version of the test. It allows the student to check his ability some days (and even hours and minutes) before the exam; Secondly, our tool is also able to be used in a *release mode*, that is, during the exam itself. In this case, page statements do not depend on the current retrieval date/time, but on a fixed seed parameter, thus being able to be easily replicated later (after the exam). The working mode changes automatically depending of the retrieval time of the test.

As it can be seen in Fig. 4, every statement question of the exams contains random strings, chosen from a huge input set of values.

A. *Generation of the input set (random values) of the questions*

The calculation of the random values necessary to give a value to the input sets of the corresponding questions is obtained as follows:

1. A single input character string is build, made up

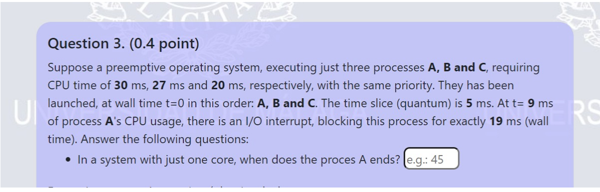

Fig. 4. Exam question statement example, with the random parameters in bold.

of the concatenation of the student's ID number, the date and exact time (in seconds) of the test page generation, the public and private IP, and a secret keyword.

2. A 160–bit string is obtained using the SHA1 algorithm (Secure Hash Algorithm (see table I). The potential number of combinations of the hash sequence is around 10^{48}.
3. Each exam (there are five of them, up to date [2]) contains around 50 input variables of different types (Boolean, integers in a range, discrete set, etc.), so we generate, at starting from arbitrary sub–strings of the hash sequence, about 50 random numbers[1]. Let's look at an example question:

> *"Convert the float number –**val1 ($rnd1)** to IEEE754, and represent the result in **val2 ($rnd2)**"*

The numbers $rnd1 and $rnd2 are 5 and 1 bit width, respectively, in the example, and they could be the first 6 bits of the SHA1 string. A specific function converts the random number into the corresponding element of the input set. Thus, in the example, if $rnd1 is 01001, the corresponding value is val1 = 10010.011 (18.375), where the middle bits is the random part. As can be seen, it is intended that the correspondence between $rnd1 and val1 generate sequences so that the questions are of similar difficulty, so that a non-random part is maintained. The result, for a certain student, at a certain moment, is

> *"Convert the float number **-18,375** to IEEE754 and represent the result in **hexadecimal**"*

B. Implementation of simulators and right–answer calculation

It is clear that, when the number of students is high, it would be unfeasible for the teacher if every question has an answer that could not be automatically calculated. In this work we have implemented around 12 simulators of the 18 existing ones in C# language (see section II-A), using the PHP language, to automatically obtain the answer for each sentence of the exam[2]. With these simulators, we do not only prepare the exam on the server-side, but also the right answer to every question.

In the *training mode*, answers can be seen by the student using a custom buttom, labeled "*unveil*", as it can be seen in Fig. 5.

Otherwise, both in the *training and release modes* allow to submit the student's answer to the database, where it is stored, close to the right answer, a time stamp, and the base random string (which adds redundancy, and it may be required for security) (Fig. 6).

Finally, a third operating mode (*feedback mode*) is incorporated in our tool. It automatically evaluates each section, according to similarity criteria automatically, established by the teacher, between given answer and right answer, and greatly facilitating the evaluation work. Since the student can send more than one answer, the evaluation chooses the last one delivered.

Regarding the possibility that students can use the simulators on which our tool is based as additional help during the exam, we understand that the ability and the time necessary to transfer specific questions to the simulators, check the results, and copy back the obtained answers to their examination test would only be possible in cases of very outstanding students.

IV. Final results and conclusions

We must be aware that, in exceptional circumstances such as those that occurred in the 2020 academic year, any statistical conclusions obtained through comparison with other courses must be handled with great care. In particular, with the data of around 109 students presented to the exam, they have passed 100% and their average score improves on the previous year by around 0.6 points, although it is impossible to determine that this improvement is due to the new training mode of the examination or no copies. What has been observed, through the, almost 4000 records in the database. is that the students have answered a mean of around four complete exams in the training mode (in addition to those carried out without delivery, using the option "reveal answer"). Regarding whether or not they have been copied during the exam, it is almost impossible to determine, but anonymous inquiries through their own classmates confess that "it was almost impossible, because all the exams were very different and there was hardly any time."

The most important aspect of the proposed evaluation method lies in the immense number of possible combinations of tests that make it practically impossible for two randomly chosen tests to show any similarity. But to value the qualities, let's see a simple

[1] While the SHA1 sequence is not, in fact, a random number, this term will be used throughout the paper for clearness

[2] It is clear that this enormous translation and implementation effort of C# objects to PHP is only rewarded if an exam is designed in a way that can be reused in many future examinations. In fact, after programming an exam for every chapter of the course, it can be consider that we have an examination tool which can be reused for decades.

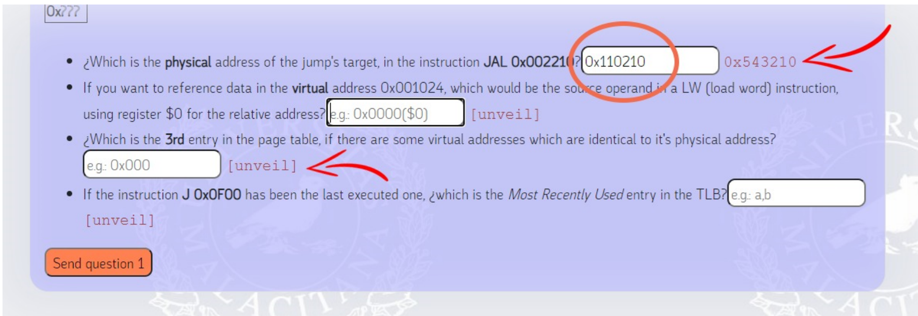

Fig. 5. Part of an exam question statement (in training mode). After each question, an element appears with the text "reveal" (red botton arrow), which, when marked with the mouse, shows the corresponding answer (red top arrow). Also note that when you fill in any field, the submit button is highlighted (in bright color).

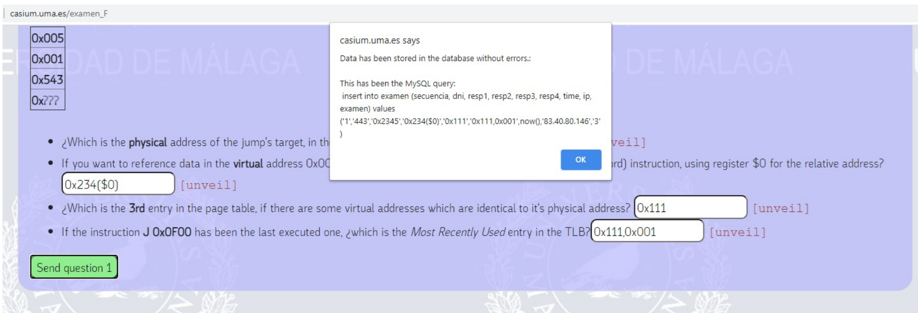

Fig. 6. Each group of 3 or 4 questions is submitted independently, and several submissions can be made, all of which are saved in a database. The green button indicates a successful submission after any change in any field. A pop-up message advises of the correct storage of the information in the database.

comparison with another scenario. We will now assume that the student is presented with 20 questions (we believe that this is a high enough number that much of the subject is covered by the questions). In the scenario offered by Moodle, 20 random questions would be selected. In each one, one of 10 questions would be selected. As advantages, there is a high number of combinations that make it very difficult for two exams to be the same. As drawbacks, there is a possible imbalance between exams, the possibility of leaving gaps, the difficulty of writing 200 questions and determining their corresponding answers. With regard to reusing the exam, it is likely that, in successive courses, students may study the 200 answers without understanding how they are solved. In our scenario, on the other hand, we would have around 20 questions with an average of 256 combinations (8 random bits) per question, whose answer is automatically calculated. Furthermore, the number of combinations can grow without increasing complexity, making it impossible to learn the answers without knowing the resolution procedure.

In short, this article shows how, through the use of specific simulators for the subject, it is possible to design an exam with countless variants that facilitate evaluation and training in complicated academic circumstances such as those we have suffered in 2020.

Acknowledges

We would like to thank all the teachers and students of the Computer Fundamentals subject for their willingness to collaborate in the development of this work, and especially those who, with their valuable suggestions, have contributed to improving the system. We also want to thank the Department of Computer Architecture of the University of Malaga, for facilitating virtual teaching in the form of financial and human resources. And of course, to the University itself, for financing this work, through the PIE-19-096 Educational Innovation Project (CASIUM), which belongs to its Own Teaching Plan.

Referencias

[1] Martin Dougiamas, "Moodle," .
[2] Luis F. Romero and G. Bandera, "The CASIUM project," 2021, https://casium.uma.es, visited 2021-05-31.
[3] G.W. Poehlein, *Universities and information technologies for instructional programmes: issues and potential impacts*, Technology Analysis & Strategic Management, Vol. 8 No. 3, pp. 283-290, 1996.
[4] T. Volery and D. Lord, *Critical success factors in online education*, International Journal of Education Management, DOI: 10.11080951354001034473, 2000.
[5] P.C. Candy, G. Crebert, and J. O'Leary, *Developing Lifelong Learners through Undergraduate Education*, National Board of Employment, Education and Training, Australian Government Publishing Service, 1994.
[6] W. A. Draves, *Teaching online. River Falls*, WI: Learning Resources Network (LERN), 2002.
[7] L.M. Medina-Herrera and J.C Miranda-Valenzuela, *What Kind of Teacher Achieves Student Engagement in a Synchronous Online Model?*, IEEE Global Engineering Education Conference (EDUCON), 2019.
[8] K. Hmieleski and M. V. Champagne, *Plugging into course evaluation*, The Technology Source, 2020.
[9] K.Morrison, *Online and paper evaluations of courses: a literature review and case study*, Educational Research and Evaluation, 19:7, 585-604, DOI: 10.1080/13803611.2013.834608, 2013.
[10] R. J. Avery, W. K. Bryant, A. Mathios, H. Kang, and D. Bell, *Electronic course evaluations: Does on online delivery system influence student evaluations?*, Journal of Economic Education, 37(1), 21–38, 2006.

[11] J. Donovan, C. Mader, and J. Shinsky, *Constructive student feedback: Online vs. traditional course evaluations*, Journal of Interactive Online Learning, 5(5), 283–296, 2006.

[12] E. Gamliel and L. Davidovitz, *Online versus traditional teaching evaluation: Mode can matter*, Assessment & Evaluation in Higher Education, 30, 581–592, 2005.

[13] W. B. Burton, A. Civitano, and P. Steiner-Grossman, *Online versus paper evaluations: differences in both quantitative and qualitative data*, J Comput High Educ. 24:58–69, DOI 10.1007s12528-012-9053-3, 2012.

[14] R. M. Carini, J. C. Hayek, G. D. Kuh, J. M. Kennedy, and J. A. Ouimet, *College students responses to web and paper surveys: Does mode matter?*, Research in Higher Education, 44(1), 1–19, 2003.

[15] J. B. Kasiar, S. L. Schroeder, and S. G. Holstad, *Comparison of traditional and web-based course evaluation processes in a required, team-taught pharmacotherapy course*, American Journal of Pharmaceutical Education, 66, 268–270, 2002.

[16] J.R. Stowell, W.E. Addison, and J.L. Smith, *Comparison of online and classroom-based student evaluations of instruction*, Assessment & Evaluation in Higher Education, Vol. 37, No. 4, 465–473, ISSN 0260-2938, Taylor & Francis, http://dx.doi.org/10.1080/02602938.2010.545869, 2012.

[17] G. Basilaia and D. Kvavadze, *Transition to Online Education in Schools during a SARS-CoV-2 Coronavirus (COVID-19) Pandemic in Georgia*, Pedagogical Research, 5(4), em0060, e-ISSN: 2468-4929, 2020.

[18] Najmul Hasan and Yukun Bao, *Impact of "e-Learning crack-up" perception on psychological distress among college students during COVID-19 pandemic: A mediating role of "fear of academic year loss"*, Child Youth Serv Rev.; 118: 105355, DOI: 10.1016/j.childyouth.2020.105355, 2020.

[19] Marcus L. George, *Effective Teaching and Examination Strategies for Undergraduate Learning During COVID-19 School Restrictions*, Journal of Educational Technology Systems, Vol. 49(1) 23–48, DOI: 10.1177/0047239520934017, 2020.

[20] ACM/IEEE-CS Joint Task Force on Computing Curricula, "Computer science curricula 2013," Tech. Rep., ACM Press and IEEE Computer Society Press, December 2013.

[21] "Microsoft Store," 2021.