

Online evidence application platform

Plataforma de aplicación de evidencias en línea

LÓPEZ-DÍAZ, Roberto Enrique†*, NIETO-YÁÑEZ, Alma Delia, HERNÁNDEZ-BÁEZ, Irma Yazmín and LEÓN-SOSA, Sandra Elizabeth

Universidad Politécnica del Estado de Morelos. Mexico.

ID 1st Author: *Roberto Enrique, López-Díaz* / **ORC ID:** 0000-0003-1716-6704, **CVU CONACYT ID:** 212691

ID 1st Co-author: *Alma Delia, Nieto-Yáñez* / **ORC ID:** 0000-0001-8554-8152, **CVU CONACYT ID:** 212730

ID 2nd Co-author: *Irma Yazmín, Hernández-Báez* / **ORC ID:** 0000-0002-3078-9618, **CVU CONACYT ID:** 212658

ID 3rd Co-author: *Sandra Elizabeth, León-Sosa* / **ORC ID:** 0000-0002-6047-4713, **CVU CONACYT ID:** 648491

DOI: 10.35429/JTAE.2021.14.5.6.11

Received: July 15, 2021; Accepted: December 30, 2021

Abstract

Due to the fact that one of the main activities carried out during the teaching of a course is the evaluation of evidence generated by students, safeguarding those evidence in an adequate way so that it can be graded later, becomes an important practice to carry out. This article addresses the development of a tool that allows the management of the evidence generated in a programming course of the Ingeniería en Informática in the Universidad Politécnica del Estado de Morelos (UPEMOR). This tool was developed under the software development model called improved cascade, using technologies that allow operation through the Internet, which would allow faster access to students' evidence, as well as keeping them better organized and accessible. In addition, this tool would allow defining the items that will be evaluated in each evidence, as well as being able to carry out the evaluation of them from the same place.

Resumen

Debido a que una de las principales actividades que se realizan durante la impartición de un curso, es la evaluación de evidencias que generan los alumnos, el resguardar dichas evidencias de forma adecuada para que posteriormente sean revisadas, se vuelve una práctica importante de realizar. Por lo anterior, en este artículo se aborda el desarrollo de una herramienta que permite la gestión de las evidencias generadas en una materia de programación del plan de estudios de la Ingeniería en Informática de la Universidad Politécnica del Estado de Morelos (UPEMOR). Dicha herramienta fue desarrollada bajo el modelo de desarrollo de software llamado cascada mejorada, usando tecnologías que permiten el funcionamiento a través de internet, lo cual posibilita tener un acceso más rápido a las evidencias de los alumnos, así como mantenerlas mejor organizadas y accesibles. Además, esta herramienta permitirá definir los rubros que serán evaluados en cada evidencia, así como poder llevar a cabo la revisión de estas desde un mismo lugar.

Platform, Evidence, Online

Plataforma, Evidencias, En línea

Citation: LÓPEZ-DÍAZ, Roberto Enrique, NIETO-YÁÑEZ, Alma Delia, HERNÁNDEZ-BÁEZ, Irma Yazmín and LEÓN-SOSA, Sandra Elizabeth. Online evidence application platform. Journal of Technology and Education. 2021. 5-14:6-11.

* Correspondence of the Author (Email: rlopezd@upemor.edu.mx)

† Researcher contributing as first author.

Introduction

Analyzing the activities that must be carried out during a university course, it can be observed that the time dedicated to the evaluations is the one that requires the most time and effort. This perspective consider that this activity is divided into several parts ranging from the design of the evaluation, the application and resolution by the students and finally the grading of the work done by them.

Currently you can find various digital platforms oriented to the administration of courses and evidences of several types, some that can be free access, if exists an institutional agreement, such as those provided in (Google, s.f.) and (Moodle, 2020), or there are also those where you pay for the service offered, which includes different tools, such as those offered in (Instructure, 2021) and (Blackboard, 2021). However, although the characteristics of the evidences that can be defined in such platforms, can be considered suitable for many courses within the curriculum of a university career, it is possible that those have limitations when is a computer programming course, where students must write codes in some programming language.

Is in this type of course where (at least in the institution), the way to generate an evaluation is carried out is to start with the teacher defining and providing the characteristics of the programs to be developed, and after, students using an integrated development environments (IDEs) that include compilers or interpreters, to write, execute and debug the codes they develop, repeating this process until the requested result is obtained. At the end, students deliver the codes using a physical medium, which is prone to damage or misplacement or a digital medium, which offers some advantages over the physical, but is also susceptible to errors or misuse.

The previously explained are reasons from which the project presented in this document was conceived, which consists of the development of a platform that allows teachers to manage the material generated for the course, define activities to perform the evaluation, as well as the respective evaluation guides with which the grading will be made.

On the other hand, students will also be provided with tools to generate the evidence (programs) that are requested, by including a development environment to write and deliver these programs to be graded later with the guides defined by the teacher.

This document has the following sections: Methodology for development of the platform, where the characteristics of the technology and methods to be used in the development are mentioned; Design of the solution, which presents the characteristics of the functionalities that the platform must have; Results, which includes the description and images of the functionalities developed and finally the conclusions that were reached with what has been developed so far.

Methodology for platform development

The methodology for software development used in this project was the improved or feedback waterfall model (Sommerville, 2005). In this model, activities are carried out in stages one after another, providing at the end of each stage, documentation generated during this stage that serves as useful information to start with the next stage. Because in the original waterfall model it is proposed that the phases be carried out one after another, without considering possible problems that would need corrections of the information generated in a stage, the improved cascade model considers allowing a back to the previous stage so that the information that is needed in the current stage is verified and corrected. Additionally, one of the requirements that make it possible to use this model is the clear definition of the requirements of the software to be developed in order to avoid having to make corrections to them. The stages of the software development model are:

1. *Requirements definition.* Information about functionalities, restrictions and objectives needed in the software, are obtained from the future users of it.
2. *System and software design.* Based on the obtained information in previous stage, a preliminary design is generated for review and acceptance. The design includes architectural pattern, user interface and database model.

3. *Implementation and Testing.* A software implementation of preliminary design is carried out and is tested. This implementation is divided in small units to reduce the functionality complexity.

4. *System integration and testing.* The developed units are integrated to be tested as a complete system. After the tests, software is delivered to final user.

5. *Operation and maintenance.* The software is launched to operation. When errors are detected, these are corrected and added to the running software.

In the following sections, a summary of the work done for the development of the project is presented.

Development of the solution

Requirements definition

Based on interviews with course coordinators, it was established that the platform should offer functionalities to manage information from remote places and without the need to install software on a computer.

In summary, the features planned to be included as part of the operation of the platform were:

- For administrators, allow the management of the information about courses, teachers and students
- For teachers, allow to manage evidence to be solved by the students; generate evaluation guides for the evidence created; visualize and grade evidence provided by the students
- For students, give them a space without distractions where to solve an evaluation and also give them a tool for compiling and executing programs.

System and software design

Considering the form of the work flow to be used in the platform, the operation of this must be via the web, so that access to it is from any point where the teacher or students are.

Therefore, the technologies considered for the implementation of the platform were:

- HTML5, language used to structure and organize information.
- PHP, language used to give internal operation from a remote server.
- JavaScript, language used to operate from events or actions performed by users.
- CSS, language used to give presentation and dynamics to the elements included in the pages of the platform.
- Google charts, technology used for the generation of information graphs.
- SQL language and MariaDB manager, used for the management of the generated information.

The elements and technologies mentioned were combined following the architectural pattern MVC (Model-View-Controller), which separates the components used for storage of the data generated by the platform, from the components used for the execution of the operating logic. The pattern also separates the mentioned components from those used for interaction with users (Bahit, 2014). A graphical representation of the architectural model is presented in Figure 1.

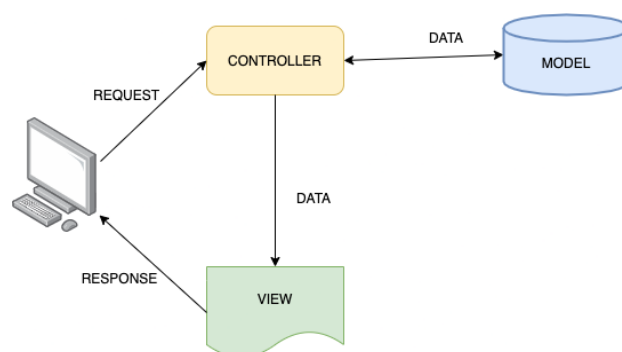


Figure 1 Architectural pattern MVC

Results

Implementation and testing

The features proposed to be developed were mostly implemented and are shown below. First, the functionalities that a teacher or course coordinator can perform are presented. Then the functionalities that a student can perform are presented.

LÓPEZ-DÍAZ, Roberto Enrique, NIETO-YÁÑEZ, Alma Delia, HERNÁNDEZ-BÁEZ, Irma Yazmín and LEÓN-SOSA, Sandra Elizabeth. Online evidence application platform. Journal of Technology and Education. 2021

Management of teachers, students and courses

The platform allows the capture, modification, consultation and elimination of information regarding teachers, the courses they teach and the students who are registered to take the programming courses included in the curriculum of the career. These functionalities can be performed by a user of type administrator or course coordinator. In Figure 2 and 3, registration of a teacher and a student respectively, is shown. In Figure 4, the consultation of teacher's course is presented.

Figure 2 Interface to register a teacher on the platform

Figure 3 Interface to register a student on the platform.

Figure 4 Interface to consult teachers and courses they teach

Management of evidence

With regard to the evidence, the teacher of a course may perform various actions with respect to the evidence that will be applied to evaluate the students, for example, create a new evidence where they can select the date of evaluation, the start and end time of the evaluation and optionally, add some digital consultation resource that can be consulted during the evaluation (See Figure 5 and Figure 6).

Figure 5 Interface to record new evidence for students

Figure 6 Interface to add digital resources for students

When new evidence is registered, the teacher will have the opportunity to generate an evaluation guide in the form of a checklist, where he can indicate the items that will be evaluated of the evidence, allowing the assignment of percentages to each item. The number of items to be evaluated can be modified, increasing or decreasing in quantity, the percentages assigned to each item and the total percentage of all items are validated. An example of this functionality can be seen in Figure 7.

| Características a cumplir | Puntaje |
|--------------------------------|---------|
| Creación de pilas *Criterio | 40 |
| Creación de colas *Criterio | 40 |
| Código comentado *Criterio | 20 |

Figure 7 Interface to register an evaluation instrument for evidence generated on the platform

Grading evidence

In addition to the previous functionalities, the teacher will be able to grade the evidence submitted by each student using the evaluation guide determined for an evidence, assigning the percentage obtained in each item, as well as being able to add observations in each category defined in the guide. In Figure 8 an example can be observed.



Figure 8 Interface for grading evidence

As can be seen in Figure 8, the teacher will have a viewer with the code written by the student, so that he can see it or copy it to the clipboard to perform some other action with it.

Generation of reports

Finally, a teacher can generate lists with the grades obtained in each evidence solved by the students in a downloadable file. See Figure 9.

Calificaciones: Estructura de datos A

Profesor: Roberto Enrique López Díaz

| Alumno | ED Introducción a ap | ED Uso de colecciones | Evidencia Pilas y Co |
|------------------------------|----------------------|-----------------------|----------------------|
| José Manuel Ferman Alcantara | 90 | 0 | 90 |

Figure 9 Example of student evidence grades report

Application of evidence

Students will be able to answer an evaluation where programs are written in a programming language. To allow this operation, a software add-on was included in the platform that allows the writing, compilation and execution of code. Figure 10 and Figure 11 present images of the functionalities described.



Figure 10 Interface for writing evidence



Figure 11 Interface with a program compilation displayed

It is worth mentioning that when a student is solving an evidence, the platform automatically saves from time to time, a backup of the code written by him, in such a way that, if any situation were presented as power outages, the progress he had is not lost and, in this way, the student could be evaluated with the developed code before the incident.

Consultation of digital resources

Students will also be able to consult digital resources shared by the teacher of the course, during the time in which the evidence is solved. Figure 12 shows the functionality mentioned.



Figure 12 Interface with consultation of a shared digital resource

Consultation of graded evidence

Additionally, students will be able to view the grades assigned to each item with which an evidence is evaluated. Figure 13 shows an example of the functionality described

| Valor del reactivo | Características a cumplir | Cumple | Puntaje obtenido | Observaciones |
|--------------------|---------------------------|--------|------------------|---------------|
| 40 | Creación de pilas | Si | 40 | Na |
| 40 | Creación de colas | Si | 40 | Na |
| 20 | Código comentando | Si | 10 | Na |

Calificación Final: 90

Figure 13 Interface where a student consults the grades of an evidence

Conclusions

In this document, the progress in the development of a platform for the application of evaluations to students of programming courses was presented.

It can be seen that the use of the improved waterfall software development model was appropriate even though such a model can be considered outside of current or agile models. It is considered that part of this result was obtained because the requirements were clearly identified from the beginning.

It can be considered that the choice of the MVC pattern and the technologies for operation in web environment, will allow developed components to have the proper maintenance and can also be updated to more recent versions.

It should be noted that the use of the software add-on for compiling and running programs occurred because the first tool selected to do so was not possible to use. Although this add-on gained a beneficial result, this technology only granted a trial period which was just enough to perform the tests of this first advance in development. A future job that is proposed is to obtain the license of said complement or use a similar technology within the platform that allows to obtain the same functionality obtained in this advance

An additional advantage obtained is that being in a situation of confinement such as at present days, this platform would allow to apply the type of evidence described in this document in a more controlled way and with the possibility that it is not necessary for students to submit their files in a physical medium or by email, since the same platform would collect the necessary information for them to be evaluated.

References

Alvizu, A. (March 27, 2016). A Devel. ORetrieved from: www.adevelca.com/blog/que-es-el-patron-mvc.

Blackboard. (January 2021). Retrieved from: <https://www.blackboard.com>.

Bahit, E. (December 2014). POO y MVC en PHP. Retrieved from: www1.herrera.unt.edu.ar/biblcet/wp-content/uploads/2014/12/eugeniabahitpooymvcenphp.pdf

Google. (s.f.). Obtén más tiempo para enseñar e inspirar a los alumnos con Classroom. Retrieved from: <https://edu.google.com/intl/es-419/products/classroom/>.

Instructure. (January 2021). Obtenido de Teaching and Learning. To the Power of Canvas. Retrieved from: <https://www.instructure.com/>.

Moodle. (November 7, 2020). Acerca de Moodle. Retrieved from: https://docs.moodle.org/all/es/Acerca_de_Moodle.

Sommerville, I. (2005). Ingeniería del software. Madrid: Pearsonn Addison Wesley