Applying PHP Codeigniter For Easy Web Development

Cristian Vidal-Silva, Claudia Jiménez, Erika Madariaga, Luis Urzúa

Abstract— Within the framework of the current advances in web technology, the main objective of the work is to describe the practical advantages of the PHP CodeIgniter 3.0 web development tool to provide a particular solution: MSRS v1.0 Manager Selection and Recruitment System, based on the requirements and structural design of MSRS and then present details of its implementation and operation. Thus, MSRS v1.0 is a web system that supports personnel selection processes in competitions under a specific professional profile, according to the definition of skills necessary for the position. Once the competencies and the level of the required domain have been defined, the items of questions to be evaluated are prepared under various reagent formats, whose responses reach a score that discriminates between the levels of the domain of the applicants, obtaining the most suitable candidate for the charge. The use of PHP CodeIgniter 3.0 stands out since, given a correct analysis of requirements and its logical database model, it allows the rapid development of web applications.

Index Terms— PHP, PHP CodeIgniter, Web Development, MSRS

1 INTRODUCTION

PHP CodeIgniter is a framework for creating PHP web applications [1] [2]. Just like its PHP base language, PHP CodeIgniter is a free and free-use product for the development of PHP applications. In this case, CodeIgniter contains a series of aids for the creation of advanced PHP applications that facilitate the progression of web applications, in addition to the definition of an orderly architecture and programming, with the inclusion of additional tools (plugins) for application development Versatile and safe [3] [4]. PHP CodeIgniter uses a standard web application development methodology with database access called Model View Controller (MVC). The model represents a software architecture that precisely segments the source code of the applications into three components:

1) Model: The model encapsulates the functions that work directly with a database so that they can be invoked from other modules of the application, so that the other modules do not directly worry about these issues with the database, but rather, they use these functionalities of the model.

2) View: A view encodes and preserves the final presentation to the user of an MVC application. That is, through a view, information and outputs are presented to the user in some defined format, not only in web format.

3) Controller: A controller is responsible for linking models and views, as well as other modules required for the generation of results.

Precisely, the workflow of a PHP CodeIgniter application involves the operation of MVC components, as illustrated in Figure 1 [3]. Figure 1 shows that any request for a web page in PHP CodeIgniter starts at index.php, a page that is hosted at the root of the site, to filter and locate that page for processing (routing). Precisely, the routing stage verifies if the required web page had already been processed and stores that info in the cache, which is configurable, to avoid repeated processing. If the required web page is not in the cache, then a security treatment is carried out, according to a previous configuration, to then give way to the controller, which communicates with a series of modules to produce the page. The controller then generates the page through appropriate views sent to the browser. Finally, the process verifies whether or not the page was in the previous cache, to enter it in case of its non-existence, optimizing its access in future requests.

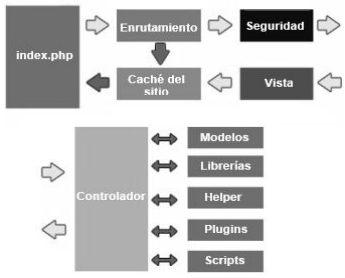

Fig. 1: Workflow of a PHP CodeIgniter Application

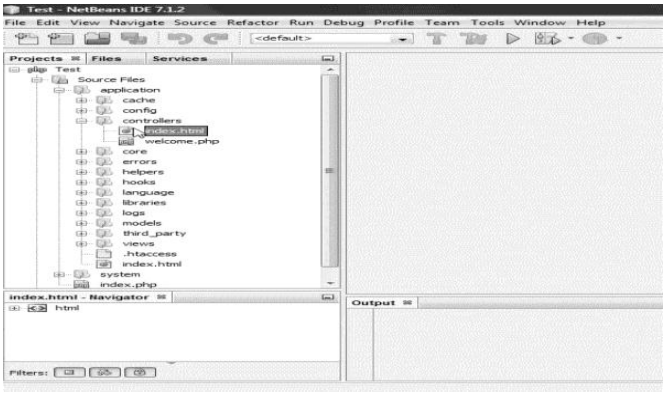

Fig. 2: Applications Folder Structure of PHP CodeIgniter solution.

- *Cristian Vidal-Silva, professor at Departamento de Administración, Facultad de Economía y Negocios, Universidad Católica del Norte, Antofagasta, Chile. E-mail: cristian.vidal@ucn.cl*
- *Claudia Jiménez, director of Ingeniería Civil Informática, Facultad de Ingeniería y Negocios, Universidad Viña del Mar, Viña del Mar, Chile. E-mail: cjimenez@uvm.cl*
- *Erika Madariaga, Facultad de Ingeniería, Ciencia y Tecnología, Universidad Bernardo O'Higgins, E-mail: erika.madariaga.garcia@gmail.com*
- *Luis Urzúa, professor at Escuela de Kinesiología, Facultad de Salud, Universidad Santo Tomás, Talca, Chile. E-mail: lurzua@santotomas.cl*

By downloading PHP CodeIgniter [1] [4], you get a PHP project that presents the main structure of an application in this framework. From a practical point of view, the NetBeans multiplatform tool [5] integrates directly with PHP applications and is used to develop PHP CodeIgniter applications. In the spatial structure of a PHP CodeIgniter solution, the main working folder is an application, where the folders for models, views, and controllers are presented to host the model, view, and controller modules of the application to be developed, as presented in Figure 2. It is important to note that in MVC applications, controllers are the intermediate modules between models and views. PHP CodeIgniter presents an object-oriented work environment for controllers and models, which are extensions of the CI_Controller and CI_Model classes, respectively. Views are modules invoked from the controllers because, in MVC, controllers are responsible for storing the applications logic and call the required views to show results. In this way, controllers decide what to do when a request is received, and views will decide how to display results. In summary, the logic of an MVC application will reside in the controller, and the view contains the appearance and design of the page to be presented to the user.

2 MSRS: MANAGEMENT SELECTION AND RECRUITMENT SYSTEM

Among the main requirements of the MSRS, it should be mentioned that it has been developed in a free web environment, without the cost of licenses, with database interaction. In this context, PHP CodeIgniter and MySQL are the chosen tools. From a functionality point of view, the main objective of the MSRS is to allow the registration (entry, update, and deletion) of Contests, for which Domains are defined. Because of this, it is necessary to identify Competences, which can belong to multiple Domains, and a Domain can contain many Competences. The measurement of Competencies is posed through the definition of numerous questions such as multiple-choice questions (Alternative Question), questions with written answers (Written Question), paired terms questions (Paired Question), and order questions (Order Question). Excepting questions with a written response, whose level of correction in the response is determined by external reviewers, sets of responses are defined, with the identification of the correct answers. Also, it is essential to define the attributes and behavior of each of these entities. Those are the base structure of the MSRS, to make effective the realization of exams for the selection of personnel.

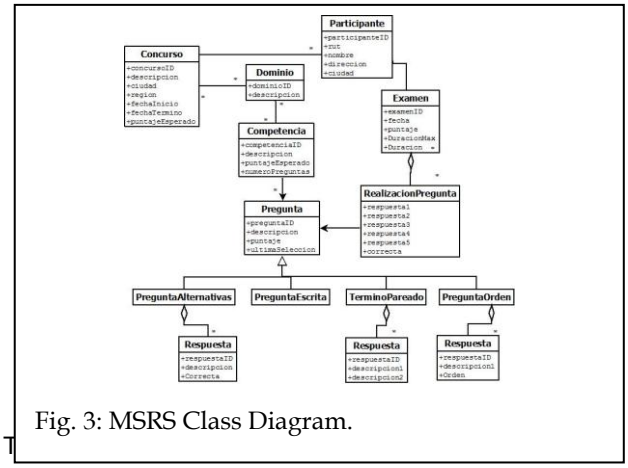

Fig. 3: MSRS Class Diagram.

The respective Contest and performs only one Exam. Each exam instance contemplates the realization of questions (Question Realization) that are associated with the question instances of the base system, to determine the integrity of the Participant's answers. As in the base system, it is required to define the attributes and behavior of these entities. Figure 3 shows the initial UML class diagram [6] of MSRS. It should be noted that the relational data model of the MSRS presents a consistency between its tables and each of the classes of the model in Figure 2, except in the Participants table where it is distinguished between Administrator and Participant users.

3 APPLYING PHP CODEIGNITER AND RESULTS

Given the MVC nature of CodeIgniter for working with Databases and the characteristics of the relational model of the MSRS, it is necessary to define model and controller classes, and the views for each of the system tables.

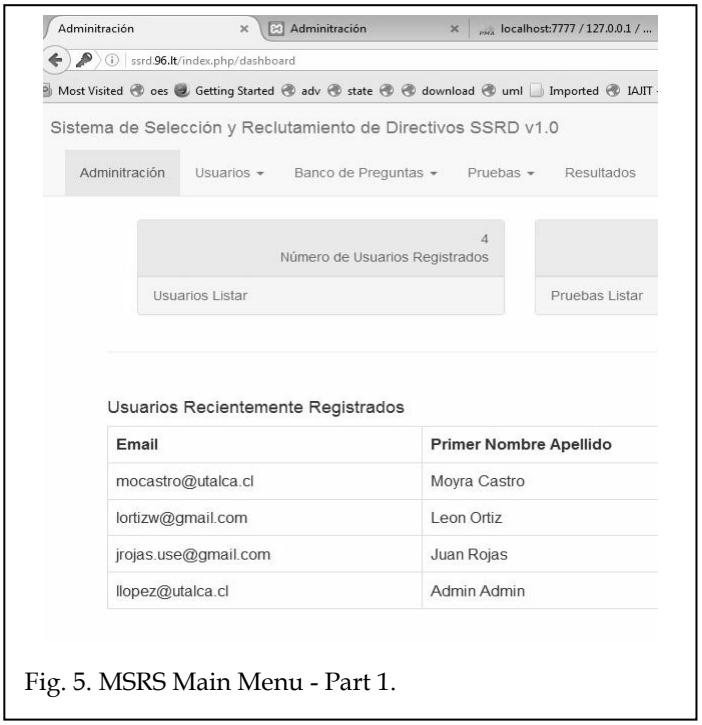

Fig. 5. MSRS Main Menu - Part 1.

As indicated by [2] and [3], PHP CodeIgniter solutions use MVC as the underlying development methodology, which requires visualizing each application functionality with that perspective. For example, to perform the MSRS login (login), an approach similar to that described in [7] is used. Figure 4 shows the login operation of the MSRS. After starting a session, Figures 5 and 6 show the main menu of the MSRS to

Fig. 4 MSRS Login.

present each of the main system options for Administrator users. Concerning non-Administrator users, the main menu displays only the options to update the account (My Account), to perform a test or test (Tests), and to obtain the results of the test (s) performed (Results).

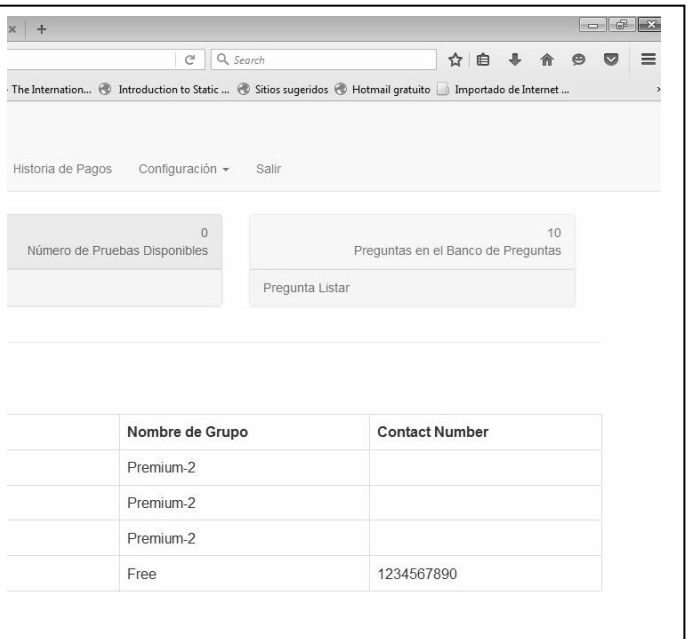

Fig. 6. MSRS Main Menu - Part 2.

4 FINAL DISCUSSION

As previously mentioned, there are two types of users in the MSRS: Administrator and Participant. Among the main functions of the first one is to feed the MSRS with the definition of competitions, examinations for said competitions, and the associated questions.

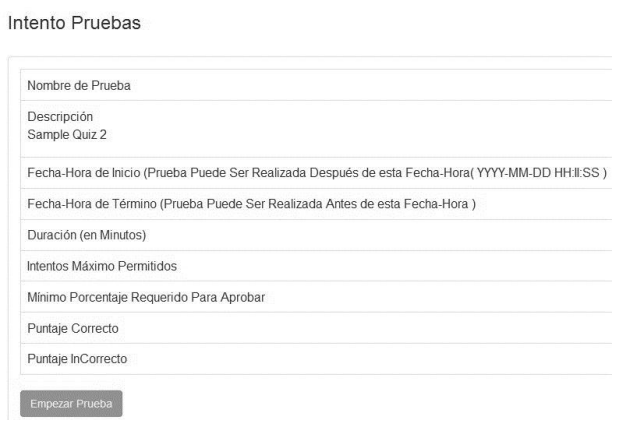

Fig. 7. MSRS Test Attempt.

Concerning Participating users, they are responsible for testing the recruitment contests. Figure 7 shows the beginning of an Exam with parameters already established by an Administrator user. The MVC framework of PHP CodeIgniter 3.0 defines a modularization of components. It allows separation of interests of each module, even when the associated classes are not “naive” as in the AOP paradigm [8] [9].In the future -in addition to implementing the MSRS, the idea of reviewing the feasibility of extending PHP CodeIgniter to support software development principles oriented to other aspects, basically an extension of MVC, is evaluated. That is, to seek the most significant possible independence between models, views, and controllers so that it is the aspects themselves that establish structural and operational relationships at defined junction points. In this way, a modeling of the aspect-oriented application is required following models proposed by Vidal et al. (2012; 2014).

5 CONCLUSIONS

MSRS shows the direct advantages of development that PHP CodeIgniter 3.0 offers; that is, an easy and ordered software construction, as well as the speed of Web applications. Once you understand the operation and structure of MVC solutions in PHP CodeIgniter, the development process itself with the analysis and design of the Web application at the Database level is a fast, agile and dynamic activity. It should be noted that numerously printed and online PHP CodeIgniter documentation exists to support the Web application development process.Another clear positive property of PHP CodeIgniter is its open-source nature, which reduces the cost of licenses for its use. Besides, given its integration and support in use with free tools such as NetBeans, its applicability for the development of Web applications is facilitated.

REFERENCES

[1] CodeIgniter, “PHP CodeIgniter”, CodeIgniter 3.0, http://www.codeigniter.com. 2020.
[2] Upton, D., “CodeIgniter for Rapid PHP Application Development”, Pack Publishing, USA, 2007, ISBN: 978-1847191748.
[3] ManualCodeIgniter, “Manual de CodeIgniter desarrolloweb.com”, CodeIgniter Documentation, http://www.izt.uam.mx/spring/wp-content/uploads/2012/07/manual-codeigniter.pdf. 2020.
[4] CodeIgniterUser, “CodeIgniter User Guide”, CodeIgniter OnLine, http://www.codeigniter.com/user_guide/. 2020.
[5] NETBEANS, “Netbeans IDE”, NetBeans for Java IDE, https://netbeans.org/. 2020.
[6] Pender, T., “UML Bible”, John Wiley & Sons, Inc., New York, USA, 2003, ISBN: 9780764526046.
[7] ILUV, “ILUV2CODE”, login-with-codeigniter-php, http://www.iluv2code.com/login-with-codeigniter-php.html. 2020.
[8] Vidal, C., R. Schmal, S. Rivero, and R. Villarroel, “Extensión del Diagrama de Secuencias UML (Lenguaje de Modelado Unificado) para el Modelado Orientado a Aspectos”, Información Tecnológica, vol. 23, no. 6, pp. 51-62, December 2012, doi: 10.4067/S0718-07642013000500002.
[9] Vidal, C., S. Rivero, L. López, and C. Pereira, “Propuesta y Aplicación de Diagramas de Clases UML JPI”, Información Tecnológica, vol. 25, no. 5, pp. 113-120, October 2014, doi: 10.4067/S0718-07642014000500016.