Model and development of an Updater for the OASis desktop system applied in the ESPOCH career secretariats

By

Hernan Dario Centeno Aulla
Systems Engineer, Information and Communication Technologies Department
Email: hernan.centeno@espoch.edu.ec
https://orcid.org/0000-0002-1837-1526

Diego Bernardo Palacios Campana
Systems Engineer, Information and Communication Technologies Department
Email: dpalacios@espoch.edu.ec
https://orcid.org/0000-0003-1085-0669

Pedro Rubén Saltos Chávez
Systems Engineer, Information and Communication Technologies Department
Email: ruben.saltos@espoch.edu.ec
https://orcid.org/0000-0003-1930-3178

Byron Ernesto Vaca Barahona
Escuela Superior Politécnica de Chimborazo (ESPOCH)
Email: byron.vaca@espoch.edu.ec
https://orcid.org/0000-0002-3622-0668

María Fernanda Heredia Moyano
Escuela Superior Politécnica de Chimborazo (ESPOCH)
Email: mariaf.heredia@espoch.edu.ec
https://orcid.org/0000-0002-0145-2098

Alex Fernando Erazo Luzuriaga
Ingeniero en Sistemas, Investigador Independiente
Email: alexerazo1407@hotmail.com
https://orcid.org/0000-0002-1089-383X

Abstract

Computer systems help to automate processes, contributing to the reduction of time, so the systems must be kept updated according to the needs or requirements that appear over time; in this context, the Polytechnic School of Chimborazo (ESPOCH) has an academic desktop system called “Orion with internet services” (OASis) which is installed in the 27 career secretariats of its 7 faculties providing various functionalities such as registration of teachers and students, enrollment, validations, class schedule assignment, etc. This academic system is subject to changes and improvements, so it needs to be updated. To carry out this process, the computer technician of each faculty is required to download the latest version and install it manually, which takes a long time to perform this task. In this study, a model and development of an Updater for the desktop system (OASis) applied in the career secretariats of the ESPOCH was carried out. To evaluate the functionality of the developed system, a form with 4 questions was applied, which were answered by the 27 career secretaries considered as the universe and representative sample of the study. Using the Chi-square statistical method with a margin of error of 5% and a degree of freedom of 6, a contingency value of 1.33 and a critical value of

12.59 were obtained, thus verifying the hypothesis proposed and fulfilling the objectives for which the system was created. Finally, the results described in this document will be used to improve efficiency in academic processes by having a desktop system updater complementing the existing one, reducing installation time when an update is found

Keywords ESPOCH, updater, desktop system, OASis system, career secretariat

Introduction

Today, one of the goals of the learning process is to form people capable of interpreting the phenomena and events occurring in a social environment (Laurus, 2008) depending on the development of new Information and Communication Technologies, which has become one of the key factors in understanding and explaining the economic, social, political and cultural transformations of the last two decades (Marqués, 2001), by which, the role played by technological innovations in the scope and direction of social and cultural changes continues to be a matter of controversy (Litwin, 2000) as they constitute an educational orientation that aims to provide answers to the knowledge society (Vera et al., 2014). In the last 10 years, the introduction of information and communication technologies (ICTs) in higher education has been a constant (López, 2007), even more so when people develop in a changing environment that demands constant updating in the mastery of knowledge and skills (Soto et al., 2009). In this sense, time has passed since the incorporation of the first computers in universities, with numerous devices becoming obsolete (Mon, 2009).

This integration and use of ICTs is part of a global trend of the knowledge society and the information society, in which schools and all educational levels are forced to participate to improve the educational processes of teaching and learning (Ancira & Gutiérrez, 2011). Thus, the traditional paradigms of teaching and learning are being modified by the integration of ICTs (Vera, 2008), stating that higher education has been evolving in recent years towards the incorporation of new systems (Coll, 2007), in such a way that radical transformation demands have been raised (Minakata, 2009) to adapt to the needs of today's society passes through the exploitation of information and communication technologies in the training processes (Salinas, 1999). Technologies widely used for the development of web applications are currently presented, but many of them force the developer to mix conceptual and presentation aspects (Silva & Mercerat, 2001) represented through computer systems play an essential role in the Information Society to meet the current demands (Pérez & López, 2007). Therefore, in software engineering, it can be observed that the most significant progress has been obtained thanks to the decomposition of a complex system into parts that are easier to handle (Quintero, 2000), so object-oriented programming is a programming language that allows the design of object-oriented applications (Izquierdo, 2007), helping to understand the programming paradigm (Sala, 2003).

In this scenario, Canós (2012) states that agile methodologies emerge as a possible answer to fill this methodological gap because they are specially oriented to small projects, taking into account that in a project, the achievement of the objectives at the end of the project is the maximum desired, but most of the time, either by poor planning or poor management of resources, it is impossible to complete the project successfully (Gallego, 2012). In this way, developing software that works has the rule to follow to make an important decision in process management (Orejuela & Rojas, 2008). Likewise, in the nineties, lightweight software development methodologies emerged that sought to reduce the probability of failure due to underestimation of costs, times and functionalities (Cadavid et al., 2013); its objective was to outline the values and principles that should allow teams to develop software quickly and responding to changes that may arise throughout the project (Letelier, 2006) being Scrum an agile and lightweight process that serves to manage and control software development (Figueroa et al., 2008).

As a result of the above, it is deduced that ICT in ESPOCH has become a tool of interest to be applied within the computer systems on which depend: students, teachers and authorities. Moreover, several computer systems exist within the institution that help automate academic activities, one of them being OASis, which is focused on providing an efficient service in relation to: Teachers can have access to class schedules, upload grade reports during the academic semester, view student lists; students can have access to pre-enrollment, view their grades, class schedules, etc. Career secretaries can also enter students and teachers into the system, enroll students, validations, career changes, among others.

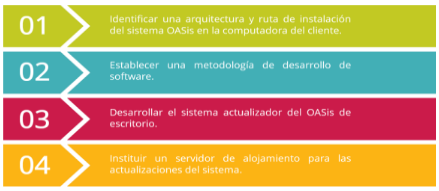
At present, OASis is available in two versions: one for the web and the other for the desktop system of the secretary of each career, developed under the .Net platform, using the IDE (Integrated Development Environment) Visual Studio 2015 together with the paradigm of OOP (Object Oriented Programming), with this it is important to note that this research aims to the desktop system, it is subject to changes and improvements, because a new version requires a technician to uninstall the previous version and install a current one.

Problematization

ESPOCH has 7 faculties and 27 careers (Table 1); in each of the faculties, a computer technician performs various functions within each faculty career. Among the activities that the computer technicians do is to support the OASis that is installed on the computers of each secretary and career director. The OASis system allows the management of academic procedures such as registration of teachers and students, enrollment, assignment of class schedules, validations, entry of academic load, and curriculum registration, among others. Academic processes vary over time; new processes appear and disappear, which is why the academic system is subject to constant changes. Therefore, in the development area of the Directorate of Information and Communication Technologies (DTIC), personnel meet the system's new requirements, periodically releasing new versions.

When there is a new version of the academic desktop system in the development area, the installers are created and sent to the IT technicians of each faculty to install the old system and install the new version. This procedure is very time-consuming, considering that each technician is in charge of more than one career. For this reason, the DTIC thought of a timely solution that would automatically keep the desktop systems of the secretaries and career directors up to date without the need to uninstall the previous system and reinstall it.

The main research objective was to model and develop an Updater for the OASis desktop system applied in the career secretaries of ESPOCH. The specific objectives of the study are shown in Figure 1.



01 Identificar una arquitectura y ruta de instalación del sistema OASis en la computadora del cliente.

02 Establecer una metodología de desarrollo de software.

03 Desarrollar el sistema actualizador del OASis de escritorio.

04 Instituir un servidor de alojamiento para las actualizaciones del sistema.

Figura 1. Objetivos específicos de estudio.

Figure 1. *Specific objectives of the study*

Table 1. *Faculties and careers of ESPOCH.*

Logo	Faculty	Careers
	Faculty of Business Administration of Business	Business Administration Finance Marketing Accounting and Auditing Transportation Management
	Faculty of Science	Chemistry Chemical Sciences Physics and Mathematics Biochemistry and Pharmacy
	Faculty of Livestock Sciences	Zootechnics Agroindustries
	Faculty of Computer Science and Electronics	Graphic Design Control and Network Electronics Electronics in Telecommunications and Networks Systems
	Faculty of Mechanics	Industrial Maintenance Mechanics Industrial Automotive
	Faculty of Natural Resources	Agronomy Tourism Forestry Renewable Natural Resources
	School of Public Health	Health Promotion Nutrition and Dietetics Medicine Gastronomy

Problematization

Study area

The research was conducted at ESPOCH, an Ecuadorian institution of higher education headquartered in the city of Riobamba, Ecuador, on the Panamericana Sur km 1 1/2; the results generated were obtained in the area of DTIC development, which provides comprehensive quality services in the areas of organizational development and information systems to the institution and external entities, using the latest technology, with trained personnel, quality standards and active and effective user participation.

Methodology

OASis is currently available in two versions: a web version used by teachers and students and a desktop version used by school directors, deans and secretariats.

Research method

For the development of this work, the inductive method allows proposing a hypothesis in order to provide an adequate solution; in this sense, it was observed that keeping the OASis desktop system updated demands much time, so with the development of the updater system, it intends to improve the experience in terms of keeping the system updated. At the same time, an evaluation of the operation of the developed system was also performed, so the following null hypothesis (H_0) and the alternative hypothesis (H_1) were proposed:

H_0: The development of an updater software will allow real-time downloading of the latest versions of the ESPOCH desktop OASis system.
H_1: The development of an updater software will not allow real-time downloading of the latest versions of the ESPOCH desktop OASis system.

Identify the architecture and installation path of the OASis system on the customer's computer

For the identification of the architecture and the installation path of the system on the client's computer, the first point considered was the physical capacity of the computer to apply the specific architecture, i.e., 32 or 64 bits to install the system. Thus, when running the installer automatically, two options are displayed: the default path (disk C) and the second to set a manual path. Thus, the challenge for the updater was to know the installation path and architecture of the operating system to proceed with the OASis system update since the updater is programmed to identify this information and proceed with the update, ensuring that the information is correct. A minimum requirement for the updater is to have an internet connection because, in the first instance, the academic system checks its current version with the web server version.

Establish the software development methodology

The development of software systems is not an easy task (Canós, 2012) which is carried out through a well-defined process where instructional design and software engineering practices are incorporated (Díaz et al., 2006). Among the functional requirements to be met by the updater is the interaction with the academic desktop system and downloading the latest version hosted on the server. Among the non-functional requirements of the updater system is its availability whenever there is an Internet connection. Based on these criteria, the SCRUM methodology was adopted for developing the computer system, as it is an agile process that can be used to manage and control complex software and product developments using iterative and incremental practices (Centeno & Santillán, 2016).

Once the new system's requirements are defined, the user tasks are defined, and the groupings of user tasks give rise to the sprint, which the development area will develop. Finally, each task performed is reviewed weekly by the coordinator and area manager (Figure 2).

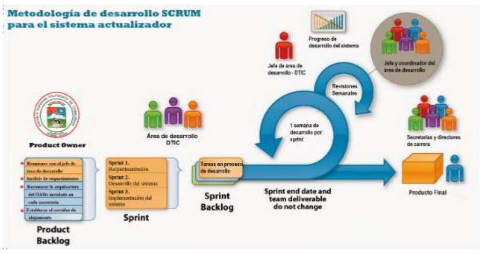

Metodología de desarrollo SCRUM para el sistema actualizador
Progreso de desarrollo del sistema
Jefe de área de desarrollo - DTIC
Jefe y coordinador del área de desarrollo
Revisiones Semanales
1 semana de desarrollo por sprint
Secretarias y directores de carrera
Área de desarrollo DTIC
Product Owner
Reuniones con el jefe de área de desarrollo
Análisis de requerimientos
Reconocer la arquitectura del OASis instalado en cada secretaría
Establecer el servidor de alojamiento
Product Backlog
Sprint 1. Requerimientos
Sprint 2. Desarrollo del sistema
Sprint 3. Implementación del sistema
Sprint
Tareas en proceso de desarrollo
Sprint Backlog
Sprint end date and team deliverable do not change
Producto Final

Figura 2. Diagrama procesos de SCRUM para el desarrollo del sistema actualizador.

During the software development, several daily meetings were held with the director and area manager to ensure compliance with the specified tasks. Finally, SCRUM was adopted as the development methodology, which is based on the following points described below:

People and roles in the project

Three people and ESPOCH were the main stakeholders

Table 2.*People and roles in the project*

ESPOCH	Product Owner	ESPOCH
Eng. Gustavo Hidalgo	Scrum Master	Director DTIC
Eng. Juan Carlos Diaz	Scrum Master	Head of development
Eng. Hernán Rye	Developer	Developer

Types of system user roles

As the end user of the developed system, it was determined that the secretaries are the only people who use the system, thus determining a single user (Table 3).

Table 3. *Types of system user roles.*

User Type	Role
Secretary	This is the user whose role is to manage the academic processes of the career to which he/she belongs through the OASis desktop system.

Tasks performed

Prior to the development of the updater system, 10 tasks were identified, which were performed during the software development; the duration of each task was estimated in hours (Table 4).

Table 4. *Tasks performed*

Person	Role	Cargo
Institution	**Name of the task**	**Hours**
	Needs analysis	
ESPOCH	Meetings with the head of the development area	
	Requirements analysis	
	Definition of physical resources	
	Set up the updater hosting server	
	System development	
	Recognize the architecture of the OASis installed in each secretariat.	
	Delete the current OASis system installed in each secretariat.	
	Download the new OASis	
	Unzip the new version of OASis	
	Delete the compressed file downloaded from the server	
	System implementation	
	Testing in a race	
	Send the installers to be installed by the technicians of each race	
	Total	

Developing the OASis desktop updater system

Table. 5 shows the tools used to develop the updater system.

Table 5. *Tools used.*

Tool	Description
Visual Studio 2015 development IDE	The .NET platform makes available a set of technologies and tools that simplify application development and proposes a solution for almost any type of application (Groussard, 2013). It is Microsoft's development interface. It consists of tools that allow developers to create applications for .NET platforms (Hugon, 2015).
Language C# programming language	It is a language that is based on C++. That is, much of the syntax and design is object-oriented. Therefore, readers familiar with C++ and OOP will find it very easy to learn how to develop applications with C# (Sierra, 2011).
SQL database engine	It is a database management system produced by Microsoft based on the relational model. Its query languages are T-SQL and ANSI SQL. Microsoft SQL Server is Microsoft's alternative to other powerful database management systems such as Oracle, PostgreSQL or MySQL (Rodriguez, 2017).
Windows Server	It provides an administrator with a complete platform at the level of AD domain management, virtualization or implementation of a cloud computing system. The environment is also tailored to the needs in order to guarantee reliability and optimal performance in complete isolation (Vera et al., 2014).

For the realization of this update system, a working algorithm was developed for its implementation as a first step. The algorithm designed meets the current requirements, which focus on identifying whether there is a new version, and then proceeds to identify the system's installation path and architecture to request the server and install the new version of the OASis desktop system (Figure 3).

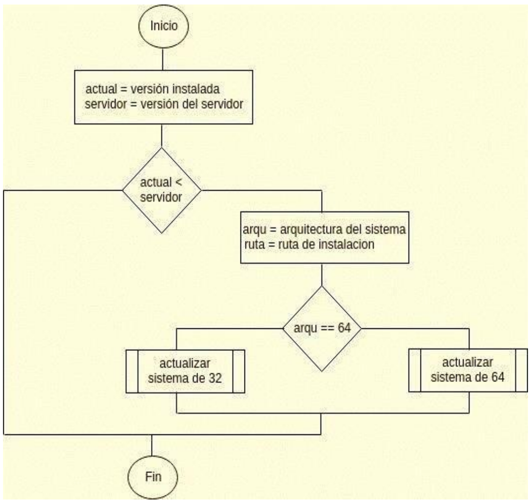

Inicio
actual = versión instalada
servidor = versión del servidor
actual < servidor
arqu = arquitectura del sistema
ruta = ruta de instalacion
arqu == 64
actualizar sistema de 32
actualizar sistema de 64
Fin

Figure 3. *Process flow diagram.*

In the first instance, the latest version of the system sent to the production server was specified in the database in order to compare it with the version running on the desktop system. In this way, when the OASis system is executed, it makes a query to the database about the current version hosted on the server, If the server version is higher, the OASis system sends the updater to run. First, the updater identifies the architecture of the client's computer it is running (32 or 64 bits), together with the path where the system is installed; once the architecture and installation path is recognized, the updater proceeds to delete all the binaries of the currently installed system and requests the server. Once the request is answered, the new system binaries are downloaded in a Zip file according to the required architecture. After finishing the download, the updater unzips the binaries, deletes the Zip file and runs the OASis system with the new version (Figure 4).

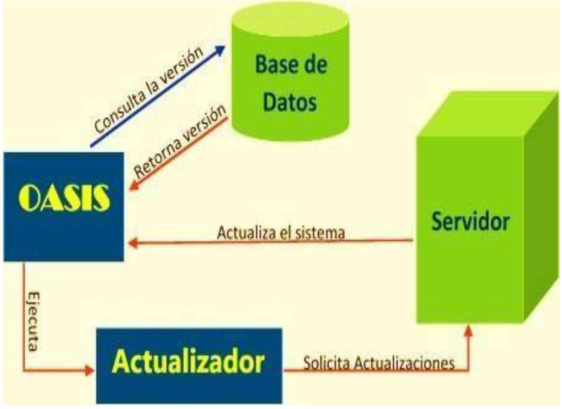


Figure 4. *Architecture of the updater system.*

Set up a web server to host system upgrades

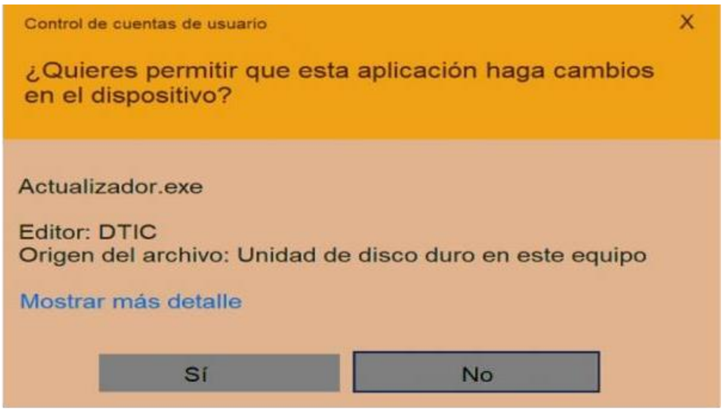
For the fulfillment of the update process, it is necessary to host in a cloud repository the binaries of the available new version. Therefore, a Windows server owned by ESPOCH was used, so it is necessary to know the server's public IP (Internet Protocol) address. Subsequently, the study proceeded to create a public folder where the binaries of the new version of the system of the two architectures (32 and 64 bits) are hosted, and this hosting file has access to the updater to download the latest versions of the desktop system.

Results

Updater installation

Within the DTIC, the development area, when a new version of OASis was identified, proceeded to generate the installers, which were sent to the computer technicians of each career so that they could be installed in the secretaries' work computers. This activity required approximately two days to complete the procedure, which could vary depending on the technician's availability. Nowadays, when a new update is identified, it is uploaded to the web server by the head of the development area. In this context, when the secretary runs her OASis system, it also runs in the background updater, whose function is to ask if there is a new version on the server and download the binaries to the secretary's machine; the estimated execution time is 1 minute.

Thus, the updater system is a desktop application that needs to be installed by a computer technician only once. With this new version presented, the system is in charge of keeping itself updated. As a result, when the OASis desktop system is executed, it queries the latest version available on the server and, if necessary, executes the updater, the one in charge of updating the system. Therefore, it is necessary to run it with administrator privileges so that the updater can overwrite the system binaries on disk (Figure 5).


Control de cuentas de usuario

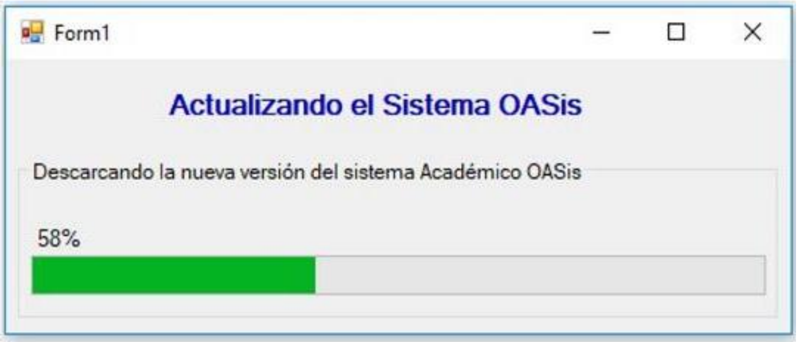
¿Quieres permitir que esta aplicación haga cambios en el dispositivo?

Actualizador.exe

Editor: DTIC
Origen del archivo: Unidad de disco duro en este equipo

Figura 5. Ejecución del sistema como administrador.

In particular, the secretary's role is to accept the system update, otherwise, the system will not proceed with the proper process, and as a consequence, the secretary will not be able to use the OASis desktop system. Once the secretary accepts the upgrade of the desktop system, the progress of the upgrade is shown on the screen, and after its completion, the OASis system is executed with the new changes, and the updater is closed (Figure 6).


Form1

Actualizando el Sistema OASis

Descarcando la nueva versión del sistema Académico OASis

58%

Figura 6. Progreso de actualización del sistema académico.
Figure 6. *Progress in the academic system update.*

The first production version was carried out through several successful development tests with the medical career's updater. This process was carried out because the system did not have some student reports. Therefore, as a first step, the new version of the system was uploaded to the server. Subsequently, the new version of OASis was installed, which includes the updater, the same that when using the OASis system, it proceeded to run in the foreground the updater identifying that it has a previous version; therefore the new version was downloaded from the server, the next step was to verify if the reports have been updated, having success in the results obtained, a product of completing the update.

Evaluation of upgrader system functionality

In this section, the study proceeded to evaluate the functionality of the updating system, taking into account four parameters: functionality, security, accuracy and behavior over time. In order to carry out this evaluation, a form was prepared with one question for each parameter, i.e., 4 questions in total, which were answered by the career secretaries who are the final users.

Population and sample

The study sample corresponds to the number of career secretaries (27).secretaries), who answered a form composed of 4 questions that were used to evaluate the functionality of the updater (Table 6).

Table 6. *Questions for system evaluation.*

No.	Evaluation criteria	Under	Medium	High
1	Do you believe the updater's use is appropriate for the OASis system?	0		
	The OASis system, after an upgrade works correctly.	0		
	Do you think that the time required for the updater to download the latest versions of OASis is appropriate?	0		
	Do you think the updater is a great help to the OASis system?	0		
	Total	0	45	

Chi-square formula

With the data obtained, the study proceeded to test the hypothesis using the chi-square statistical method, which indicates that it is the observed sum minus the expected distribution squared over the expected distribution, which is expressed with the following formula:

$$x^2 = \sum \frac{(fo - fe)^2}{fe} \qquad \textbf{(1)}$$

If the chi-square value exceeds the critical value, the null hypothesis is rejected; otherwise, if the chi-square value is lower than the critical value, the null hypothesis is accepted.

Hypotheses and Variables

The following variables were determined based on the hypotheses outlined in point 3.2.1, which corresponds to the research method:

- Dependent variable: The development of an updater software.
- Dependent variable: will allow real-time downloading of the latest versions of the ESPOCH desktop OASis system.

Degree of freedom

The calculation of the degree of freedom was defined with the following formula:

gl = (f - 1) (c - 1) **(2)**

From the questions posed, form a matrix of 3 rows and 4 columns:

gl = (3-1) (4-1) **gl** = 6

Significance level

To perform the hypothesis test, we worked with a common level of 1%, which means there is a 99% probability that the null hypothesis is accepted (Figure 7).

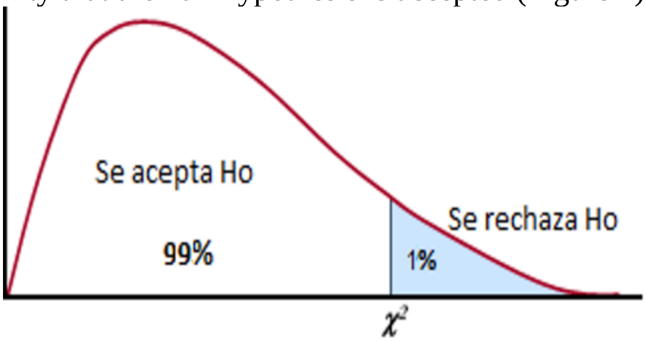


Figura 7. Nivel de significancia.

Hypothesis testing

The data obtained from the cards answered by the career secretaries are added together for each question, this result is known as the observed frequency, and the values of the observed frequency are used to calculate the expected frequency of the same. These values are expressed in Tables 7 and 8.

Table 7. *Observed Frequency*

Under	Medium	High
0		
0		
0		
0		

Table 8. *Expected Frequency*

Under	Medium	High
0	11.25	15.75
0	11.25	15.75
0	11.25	15.75
0	11.25	15.75

With the expected frequency and the observed frequency, we proceed to obtain the chi-square contingency matrix with the following formula:

$$^{2}x = \sum \frac{(fo - fe)^{2}}{fe} \qquad \textbf{(3)}$$

Whose value is shown in Table 9?

Table 9. Chi-square contingency matrix

15,75	0,035
15,75	0,321
15,75	0,003
15,75	0,194
Total	1,333

fo	faith	(fo - fe)2 / fe
	11,25	0,050
	11,25	0,450
	11,25	0,005
	11,25	0,272

Table 10 shows the chi-square distribution, where the value corresponding to the degree of freedom obtained with the established margin of error was located.

Table 10. *Chi-square contingency matrix.*

Criteria from freedom	Probability										
	0.99	0.98	0.975	0.95	0.90	0.80	0.50	0.20	0.10	0.05	0.025
1	0.0022	0.0006	0.0010	0.0039	0.0158	0.0642	0.4549	1.6424	2.7055	3.8415	5.0293
	0.0201	0.0404	0.0506	0.1026	.02107	0.4463	1.3863	3.2189	4.5062	5.9915	7.3778
	0.1148	0.1848	0.2158	0.5318	0.5844	1.0052	2.3660	4.6416	6.2514	7.8147	9.3484
	0.2971	0.4294	0.4844	0.7107	1.0636	1.6488	3.3567	5.9886	7.7794	9.4877	11.1433
5	0.5543	0.7519	0.8312	1.1455	1.6103	2.3425	4.3515	7.2893	9.2363	11.0705	12.8325
	0.8721	1.1344	1.2373	1.6354	2.2041	3.0701	5.3481	8.5581	10.6446	12.5916	14.4494
	1.2390	1.5643	1.6899	2.1673	2.8331	3.8223	6.3458	9.8032	12.0170	14.0671	16.0128
	1.6465	2.0325	2.1797	2.7326	3.4895	4.5936	7.3441	11.0301	13.3616	15.5073	17.5345
	2.0879	2.5324	2.7004	3.3251	4.1682	5.3801	8.3428	12.2421	14.6837	16.9190	19.0228
	2.5582	3.0591	3.2470	3.9403	4.8652	6.1791	9.3418	13.4420	15.9872	18.3070	20.4832

Note. Retrieved from: “Asociación de variables cualitativas: test de Chi-square” (Pita Fernández & Pértega Díaz, 2004).

Since x2 = 1.33 is less than the critical value Vc = 12.59, with a significance level of 1% and with a degree of freedom of 6, it can be concluded that the null hypothesis is accepted and the alternative hypothesis is rejected, which is "The development of an updater software will allow downloading in real time the latest versions of the OASis desktop system of the ESPOC."

Conclusions

With the updater system, it was possible to identify the architecture and route where the installation of each OASis desktop system of the career secretariat that corresponds to the same version of the operating system in function; it is so the version of the desktop system is downloaded the latest updated version of the system; also the configuration of the web server was intranet where the binaries of the latest version of the OASis system are hosted, these files are compressed in Zip format, the server has two directories that correspond to the two architectures of the system that are of 32 and 64 bits, the same one that responds to a requested download request.

This system becomes a contribution to the improvement of processes, in this case of academic type, with which it contributes to the improvement of times in activities that were previously considered for installing new system updates.

The system now identifies the updates and can quickly and efficiently perform them in real-time, reducing the installation time of the updates, directly benefiting teachers, students and secretaries of ESPOCH.

The system was installed on 27 computers corresponding to all the career secretaries of ESPOCH. Finally, using the chi-square statistical method with a margin of error of 5% and a degree of freedom of 6, it obtained a contingency value of 1.33 and a critical value of 12.59, which rejects the alternative hypothesis and approves the null hypothesis that says "the development of an updater software will allow downloading in real time the latest versions of the OASis desktop system of the ESPOC".

Conflict of Interest

It is freely and voluntarily expressed that in the development of this article, there are no personal conflicts of interest on the part of any of the authors, and therefore, the publication of the study is authorized.

Acknowledgments

As a contribution to the academic/scientific development of the institution, this research was carried out in the area of DTIC development, for which we thank ESPOCH for the technological and human support for fulfilling the objectives outlined.

References

Ancira, A. & Gutiérrez, F. (2011). Integración y apropiación de las TIC en los profesores y los alumnos de educación media superior. Apertura. 3(1): p. 142-155.

Cadavid, A., Martínez, J., & Vélez, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. Prospectiva. 11(2): p. 30-39.

Canós, J. & C. (2012).Letelier, Metodologías ágiles en el desarrollo de software.

Centeno Aulla, H. D., & Santillán Chagñay, A. T. (2016). Sistema de comunicación móvil basado enla metodología aumentativa alternativa en realidad aumentada para niños con síndrome de down (Bachelor's thesis, Escuela Superior Politécnica de Chimborazo).

Salvador, C. C., Villach, M. J. R., Saíz, R. M. M., & Llanos, M. N. (2007). Evaluación continua y ayuda al aprendizaje. Análisis de una experiencia de innovación en educación superior con apoyo delas TIC. Electronic Journal of Research in Educational Psychology, 5(3), 783-804.

Díaz-Antón, M. G., Pérez, M., Grimmán, A., & Mendoza,

L. (2006). Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica. Univ. Simón Bolívar, Caracas, Venez, 1, 91.

Fernández, G., Garijo, M., Iglesias, C., Pastor, E., Villagrá, V., Berrocal, J., Costilla, C., García, M.,& Seoane, J. (2004). Programa ingeniería de sistemas telemáticos Código. 2004.

Figueroa, R., Solís, C & Cabrera, A. (2008). Metodologíastradicionales vs. metodologías ágiles. UniversidadTécnica Particular de Loja, Escuela de Ciencias de la Computación.

Gallego, M. (2012). Metodología Scrum. Universitat Oberta de Catalunya.

Groussard, T. (2013). Visual Basic (VB. NET): Los fundamentos del lenguaje-Desarrollar con VisualStudio 2012. 2013: Ediciones ENI.

Hugon, J. (2015). C# 6: Desarrollo aplicaciones Windowscon Visual Studio 2015: Ediciones ENI.

Izquierdo, L. (2007). Introducción a la programaciónorientada a objetos. Publicación on line.

Laurus, C. (2008). Aprendizaje basado en proyectos colaborativos. Una experiencia en educacion.

Letelier, P. (2006). Metodologías ágiles para el desarrollode software: eXtreme Programming (XP).

Litwin, E. (2000). Tecnología educativa políticas, historia,propuesta. Ediciones Paidós SA.

López, M. (2007). Uso de las TIC en la educación superiorde México. Un estudio de caso. Apertura. 7(7).

Marqués, P. (2001). Algunas notas sobre el impacto de lasTIC en la universidad. Educar. (28): p. 083-98.

Minakata, A. (2009). Gestión del conocimiento eneducación y transformación de la escuela: Notas para un campo en construcción. Sinéctica. (32): p.17-19.

Mon, F. (2009). Bolonia y las TIC: de la docencia 1.0 al aprendizaje 2.0. La Cuestión Universitaria.(5): p.6.

Pérez, Y., & López, L. (2007). Multiparadigma en laenseñanza de la programación. IN IX Workshop deInvestigadores en Ciencias de la Computación.

Pita Fernández, S., & Pértega Díaz, S. (2004). Asociaciónde variables cualitativas: test de Chi-cuadrado. 1,

Retrieved fromhttps://www.fisterra.com/mbe/investiga/chi/chi.p df

Quintero, A. (2000). Visión general de la programación orientada a aspectos. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla.

Rodríguez, H. (2017). Implementación de un sistema de información para empresa agroindustrial Santa María en la ciudad de Lima.

Sala, J. (2003). Introducción a la programación. Teoría y práctica. Vol. 3.

Salinas, J. (1999). ¿Qué se entiende por una institución deeducación superior flexible. in Comunicación presentada en” Congreso Edutec.

Sierra, J. (2011). Microsoft C#. Curso de programación.

Vol. 7: Grupo Editorial RA-MA.

Silva, D. & Mercerat, B. (2001). Construyendo aplicaciones web con una metodología de diseño orientada a objetos. Revista Colombiana deComputación–RCC. 2(2).

Soto, C., Senra, A & Neira, C. (2009). Ventajas del uso de las TICs en el proceso de enseñanzaaprendizaje desde la óptica de los docentes universitarios españoles. EDUTEC. Revista electrónica de Tecnología educativa, (29).

Vera, F. (2008). La modalidad blended-learning en la educación superior. Rancagua, Chile.

Vera, J., Torres, E., & Martínez, E. (2014). Evaluación de competencias básicas en tic en docentes de educación superior en México. Píxel-Bit. Revista de Medios y Educación, (44).