An Approach for Migrating Legacy Applications to Mobile Interfaces

Viviana Cajas[1,3(✉)], Matías Urbieta[1,2], Yves Rybarczyk[4], Gustavo Rossi[1,2], César Guevara[5]

[1] LIFIA, Facultad de Informática, Universidad Nacional de La Plata, La Plata, Argentina
[matias.urbieta, gustavo.rossi]@lifia.info.unlp.edu.ar
[2] CONICET, Buenos Aires, Argentina
[3] Facultad de Ciencias Administrativas y Económicas, Universidad Tecnológica Indoamérica, Quito, Ecuador
vivianacajas@uti.edu.ec
[4] Intelligent & Interactive Systems Lab (SI^2 Lab), Universidad de Las Américas, Quito, Ecuador
yves.rybarczyk@udla.edu.ec
[5] Centro de Investigación en Mecatrónica y Sistemas Interactivos (MIST), Universidad Tecnológica Indoamérica, Quito, Ecuador
cesarguevara@uti.edu.ec

Abstract. Mobile applications changed unexpectedly people life and business models around the world. Nevertheless, there are old applications, called legacies, without adaptation to mobile devices, because this adaptation or migration have a considerable cost in dependence of software scope. Currently, most users bring constantly their smartphones and other devices with them, especially millennials. For that reason, some approaches try to solve this portabilization, generating certain improvements. However, in the majority of these solutions there is not a direct participation of users; do not consider their visual identity, analysis of feeling or mining of opinion. This paper proposes getting the behavior web application model with Markov heuristics from the widgets closeness matrix, prior to adaptation in order to include the user logic.

Keywords: legacy adaptation, Markov chains, millennials, mobile devices.

1. INTRODUCTION

By the year 2025, millennials [1] will constitute 75% of the world workforce. They are people who seek a balance between the personal and professional life, giving less importance to money, being their priority happiness in any situation; they are digital natives. For this reason, the Internet is their main tool, spending about 7 hours a day, their social networks are their main way of communicating, and they carry their mobile devices - smartphones, tablets and laptops - to all parties, even at the time of sleep, place them next to them [1]. Starting from this premise, the companies that have not adapted their business models to mobile devices run the risk of failing to gain the fidelity of users in the medium term. The new business models can be implemented

through mobile applications [2] because, in this way, they can reach a wider range of business users and customers who mostly access Web apps from their mobile devices any moment and anywhere. Unfortunately, the applications created prior to the development of mobile devices, which are known as legacy applications [3], were not designed with the foresight to adapt automatically. The legacy system is an application that has become outdated but is still used by the users (usually an organization or company) and is not wanted to, cannot be replaced, or updated easily. Legacy application usability problems have been studied in different works [4] pointing out the usability issues arisen when the App is run in a mobile device. Some of them are: the legibility of the typography, the icons, the size, orientation of the screen, etc. Also, usability on mobile applications suggests that aesthetics graphics (balance between the colors, shapes, language, music or animation) is an important concept when evaluating the overall them [5]. In [6] the authors conducted an experiment assessing the productivity and performance of users accessing both a legacy and a mobile-friendly version of an application. The results of a controlled experiment based on two sites showed usability issues requiring more scrolling and zoom in/out events than the mobile-friendly version. Moreover, [7] collected usability experiments from 10 popular applications, where 3,575 users rated the usability using the System Usability Scale (SUS) questionnaire [8]. The average SUS rating was 77.7 out of 100, which is comparable to a C grade in the university grading scale [9]. It suggests that although mobile applications have gained a reputation as usable, they are not perfectly usable. Unfortunately, due to the lack of awareness of these issues, the users are forced to interact with applications even if they have a poor usability. Moreover, the industry tried to improve usability by increasing the width of the devices.

This work proposes to use Markov chains to analyse how the application features should be designed before a migration or portabilization from legacy and what should be omitted in the mobile application. The states of the Markov chain are formed with the screens and the controls that from here on will be generalized with the name of widgets. This model helps to understand the mechanics behind navigation. Markov chains calculation is a stochastic process where the probabilities describe the way in which the process will evolve in the future. These chains correspond to a standard mathematical technique recognized for the modeling process from the definitions of Cook and Wolf models [10], and for the development of Whittaker and Poore test cases [11]. They have also been defined as a model of the consumers that seeks to describe and predict their behaviour [12]. This model also allows being able to obtain interfaces for mobile devices. The remaining of the article is organized into eight sections. Section 2 defines the Background. Related work is described in section 3. Section 4 explains the research methodology. The Markov approach for user interface design is explained in section 5. An example is provided in section 6. Results are shown in section 7. Finally, the conclusions and future work are contemplated in section 8.

2. BACKGROUND

2.1. Markov Chains

They are a type of dynamic Bayesian networks [13] that predict the state of a system at a given time from the preceding state. The most important elements for the establishment of the chain are: the state space, the transition matrix, and the initial distribution [14]. The state transition diagram graphically presents the same

information provided by the transition matrix, like in Fig. 1. The nodes (circles) represent the possible states, while the arrows show the transitions (the option to return to the same state is included). A transition probability defines the probability to move from one state to another (arrows in Fig. 1). Successful algorithms such as PageRank, created by Google, allow to grant a numerical value to each web page and from it, establish the order in which they appear after a search, use Markov chains. Proving that they are suitable for establishing web browsing models.
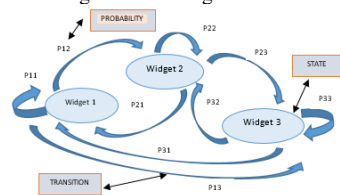

Fig. 1. Navigation Markov Chain Example

2.2. Systematic Layout Planning

This technique was proposed by Muther [15] whose acronym is SLP, allows the organization or distribution of the manufacturing plant improves productivity in the operations management through the reduction of movements, according to the flow of information or materials and the closeness the common areas to accomplish a process. In order to represent the relationships existing, a relational table of activities or areas is used, called closeness matrix. It is usual to express these needs through a code of letters, following a scale that decreases with the order of the five vowels, as shown in Table 1. This work makes an analogy between the software application and a manufacturing plant, which corresponds to space within the screen. Then, the distribution of elements in which users navigate must be optimal with respect to this matrix. And it must be considered according to the workflow and the distribution of the process. The efficient use of space within the screen is ensured according to the principle of optimal flow of the project.

Table 1. Closeness or Proximity Matrix

Key	Closeness	Value	Key	Closenees	Value
A	Absolutely necessary	5	O	Ordinary	2
E	Especially important	4	U	Without importance	1
I	Important	3	X	Undesirable	0

3. RELATED WORK

The main related studies are summarized in Table 2. The works demonstrate that Markov chains can be a useful solution in software engineering. Nevertheless, the potentially of such a technique is still underutilized in migration processes of web to mobile applications. For this reason, we propose to use Markov to obtain in the first instance the behavior model of the web application that allows generating a more efficient adaptation based on usability and accessibility for end users, taking into account the probability determined by the most used areas, common activities, as well as, the most navigated links.

Table 2. Main Characteristics of Related Works

Author	Evaluation/Future Work
Thimbleby et al. [14]	A tool to integrate Markov models in the design of a device with buttons (mobile, vending machines, recorders) is proposed and can be represented as finite state machines. The approach can be applied to abstract designs, prototypes, and animations, or fully operational systems.
Mao et al. [16]	The authors propose an extended model (EMM) to develop plans and test methods from the components to the analysis of the system. The results of the test can gradually improve the EMM to perform regression tests and can also be used to correct programs, being a semiautomatic framework.
Yanchun et al. [17]	The behavior model of a Mashup application is obtained, through a discrete Markov chain to build the model of the activities prior to building the security risk model.
Chohan et al. [18]	A hierarchical Markov chain is used to perform the software product line tests. Three models are implemented: (i) captures the potential behavior of the products; (ii) keeps record the functions with the similarities or variations between the products and (iii) makes a specification of the mapping of elements in the test transitions.
Nwobi-Okoye et al. [19]	The theoretical model of games and the concept of flow theory are used to obtain an optimal user interface modeled through chains of Markov. The validation was out three case studies; a web application, a desktop application and the comparison of mobile interfaces, leaving Windows Phone 7 at a disadvantage with respect to Android and iPhone.
Cajas et al. [20].	A systematic mapping about the portability of legacies through a migration with different strategies is developed. (i) The DOM is modified, so that the design adjusts to the size of the screens, providing an improvement in the web appearance. (ii) Model-driven development facilitates the generation of code semi-automatically from models. (iii) Mashups and (iv) middleware require high know how in order to make merge APIs and separate data sources into a single integrated interface. (v) Augmentation technique, where the user is part of the process and can select certain customization of the applications for their convenience. However, these approaches, can be used to migrate an app to a mobile version but not fully support the portabilization problem presented in this work.

The hypotheses the Markovian model contemplates are the following: i) assuming a finite number of states to describe the dynamic behavior of the widgets; ii) assuming a known distribution of initial probabilities, which reflects either what state belongs an application widget, or the percentages of widgets in each state in the application; iii) assuming that the transition from one current state to another in the future depends only on the current state (Markovian property); and iv) that the probability of this transition being independent of the time stage considered (stationary property), that is, it does not change in the study time of the system.

4. RESEARCH METHODOLOGY

This study is based on the engineering method (evolutionary paradigm) [21]: observe existing solutions, propose better solutions, build or develop, measure and analyze, repeat until no further improvements are possible. For this reason, the authors began the investigation process with a Systematic Mapping [20] that studies the problem approaches, and challenges present when migrating legacy Web applications to mobile platforms in the last decade. After examining the previously published studies, the article establishes that the Markov has not been proposed as a tool for configuring User

Deleted: the research gaps

interfaces in the migration process of legacy applications to mobile. Having said that, we present an approach based on the Markovian model. In the following section, we introduce the approach which later is instantiated in a study case. Finally we conducted a simple evaluation reporting preliminary evidence about the benefit of our approach.

5. PROPOSED APPROACH

The approach aims at adapting legacy sites by modifying its structure, content, look, and feel to become a mobile friendly app. Fig. 2 shows the process of adaptation. The different steps are described in the next sections.
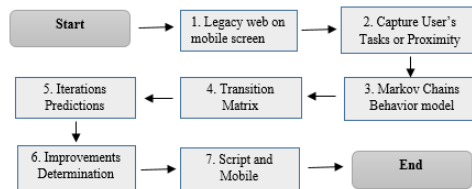

Fig. 2. Site Adaptation Process

5.1. Identify the legacy web site

A legacy website is not properly rendered in small screens. To identify it, users can minimize or modify the size of the screen to verify if the content is rearranged in some way so that they can check whether or not the content of the layout is properly rendered. Also, they can access the site from different mobile devices, and the site is not mobile friendly when it is abusively required to perform scroll or zoom.

5.2. Define expert users

The selected millennial users [1] must have certain experience within the site, in order to get the simulation for expert user probabilities of navigation.

5.3. Behavior Model Development

This process must be made to each user role because the navigation cannot be general, each one has its widgets that may or may not coincide in certain cases. The first step to get the Markov model is to fill the state spaces, as in (1) defined with the widgets on the site. Then, the initial matrix (also called the initial probability vector) is filled in as in (2). Then, it is necessary to verify that P(0) adds 1. The first time, the user always access the main page of the application. For this reason, the probabilities are 1 in the widget1.

$E = \{Widget1, Widget2, Widget3, .., Widgetn\}$ (1)
$P(0) = \{1,0,0,..,0\}$ (2)

5.4. Transition Matrix

The elements of the matrix of transition probabilities are non-negative, and the sum of the cells belonging to a row must result in 1. The order of transition matrix must be equals to the number of widgets (#Widget), so the algorithm can multiply the transition matrix by the initial matrix. For each widget discovered in Step 2 shown in a page, a

row, and a column are placed in the matrix. Then, the matrix is filled in with the probabilities of navigation. In this case to fill in the matrix is to weigh (based on a Likert scale) the values according to the related activities, in a similar way as the closeness matrix of an SLP.

5.5. Iterations Model

Once both the initial matrix and the transition matrix are obtained, the algorithm multiplies N times until obtaining a stationary matrix that represents the behavior model. It is called stationary distribution when the initial distribution does not change when it is multiplied by the transition matrix. As long as the time goes by, it does not change with the passage of time and therefore it is called stationary or invariable distribution. It is said that a Markov Chain in discrete time admits a stationary distribution to the extent that long-term probabilities exist and is independent of the initial distribution $P(0)$.

5.6. Improvements Determination

After calculating the stationary matrix, the probabilities for optimal site navigation should be recorded in the database. In addition, the strategies to establish the specific improvement of the site must be determined. Then, to make an initial improvement, first the algorithm must select the widgets related to 'A' and place them as close as possible on the user interface. On the contrary, the widgets related to 'X' or 'U' must be placed as far as possible or kept invisible and only displayed on request, because there is no direct relationship between them.

5.7. Evaluation of Distribution Alternatives

The Table 3 shows the scheme of the evaluation of the usability to should be implemented to proceed with the improvement of the user interface. The effect of the manipulation of the independent variable (distribution of the widgets) is reflected in the dependent variables.

Table 3. Evaluation Matrix [23]

Variable	Measure	Explanation
Independent	Widgets Distribution	Screen distribution affects the distance required to perform the operations and therefore the cost and efficiency of the operation or task.
Dependent	Task completion time (seconds)	The duration of tasks or parts of tasks, the time users spend in a particular mode of interaction
	Input rate	Rate of input by the user, for example using keyboard or sliders.
	Use frequency	Number of keystrokes; number of mouse clicks; number of interface actions; amount of mouse activity; scroll and zoom.

6. Running Example

Step 1. The Academic Management System (SGA) of the Indoamérica University [24] is used as running example. This platform allows the administration of teacher assignments, subject planning, class attendance, and register qualifications according to the regulation of the University. The cell phone version of the SGA displays unnecessary information, which can be cut off or incomplete, because there is not

dynamic sizing. In consequences, the users must perform scrolling, and zoom in/zoom out to accomplish the tasks. In addition, the widgets are simply displayed in alphabetic order.

Step 2. The selected millennials teachers participate with the below proceeding:
1. List the SGA widgets of the teacher role, (Table 4).

Table 4. Tested Widgets

No.	Widget Name	No.	Widget Name	No.	Widget Name	No.	Widget Name
1	Profile	5	Period	9	Attendance	13	Mail
2	Account	6	Photo	10	Teacher Self- Assessment	14	Teacher Evaluation
3	Password	7	Search	11	Calendar	15	Schedule
4	Quit	8	Open Classes	12	Grades		

2. Users in consensus must relate the widgets of Table 4 according Muther, as shown in Table 5.

Table 5. Sga Closeness or Proximity Matrix

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Profile	1		5	4	3	2	1	3	4	3	1	1	2	3	1	2
Account	2			4	3	2	4	3	4	3	1	1	3	4	1	3
Password	3				3	1	1	3	2	2	1	1	2	3	1	2
Quit	4					3	3	3	3	3	3	3	3	3	3	3
Period	5						1	3	2	2	1	1	2	2	2	2
Photo	6							3	1	1	1	1	2	2	1	1
Search	7								2	2	2	2	2	2	2	2
Open Classes	8									1	2	1	2	2	1	2
Attendance	9										1	2	2	2	1	2
Teacher Self-Assessment	10											3	1	2	2	2
Calendar	11												3	2	2	2
Grades	12													2	1	2
Mail	13														3	3
Teacher Evaluation	14															2
Schedule	15															

Step 3. The space of states is establishes to determine the behavior model, such as:

E= {Profile, Account, Password, Quit, Period, Photo, Search, OpenClasses, Attendance, TeacherSelf-Assessment, Calendar, Grades, Mail, TeacherEvaluation, Schedule }

Next, the algorithm must establish the initial matrix distribution:

P= {1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Step 4. The transition matrix is filled in with the percentages of closeness matrix. To do so, the algorithm must add the weights of the relation scale of each pair of widgets, which corresponds to the population of each row to obtain the percentages.

Step 5. In order to make the iteration model, the algorithm must multiply the initial matrix by the transition matrix to get the stationary matrix.

Step 6. The determination of improvements is made through the analysis of screen size: (i) Reorganizing the menu according to the results obtained; (ii) Including shortcuts to the features most likely to be used; (iii) Decreasing the zoom and scroll within the pages. For an initial improvement: the algorithm groups the widgets according to nearness matrix priority as represented in Table 6. This operation consists of locating the widgets with value 'A', like Profile – My Account together; then identifying the widgets with value 'E', Profile – Opening Classes; and so on.

Table 6. Widgets Profile Pair Priority

Key	Widgets Pair	Key	Widgets Pair
A	1-2;	O	1-4; 1-11;1-14;1-16;1-20;1-23;1-24;
E	1-3;1-7;	U	1-5; 1-9;1-10;1-13;1-17;1-19;
I	1-6;1-8;1-12;1-15;1-18;1-21;1-22;		

Step 7. Once raised the Markov distribution, it is necessary to determinate if the TAM of the user is improved or still require other modifications. The evaluation consists of measuring the efficiency of new widgets distribution in comparison to the old one.

7. RESULTS

Fig. 4 shows the new organization of the widgets based on the criterion of the users obtained through the closeness matrix and with mockups.

Fig. 4. Teacher Role SGA New Distribution

In order to evaluate the new user interface, we conducted a preliminary evaluation that involved ten teachers who were divided into two independent groups, control and treatment, of 5 subjects each according to Nielsen recommendation. Each subject was required to complete four tasks: (i) Locate information about their schedules, (ii) Search the planning widget, (iii) Search the e-mail inbox, and (iv) Locate the grades widget. Five participants performed the tasks on the SGA application (control group) and the five others on the SGA Mockup (treatment group). The hypothesis was that the user performance (e.g., completion time to carry out a task) should be better in the treatment group than the control group. The evaluation protocol was executed in the same way by all the subjects. They received a quick explanation of the SGA functionalities by a moderator. They were teachers who have never used the SGA application, and, moreover, they did not collaborate in design closeness matrix. In addition, two witnesses observed the behavior of subjects while performing the required tasks in their mobile device using a Chrome browser. During the assessment, the task completion time and the user interaction events count (zoom in/out, scroll, and click) metrics were evaluated. The moderator captured the time spent to complete the task as well as the required events. These values were fulfilled in a form similar to the Table 7. To evaluate the new SGA distribution a mobile application was created in Mobincube [25] that

allowed to quickly replicate the new distribution obtained. The QR code of this application was shared with each participant to install and interact with the requested tasks.

Table. 7. Old Sga Measures Vs. New Sga Measures

User	Old Sga Measures Task completion time seconds				UI Events				User	New Sga Measures Task completion time /seconds				UI Events			
	T1	T2	T3	T4	T1	T2	T3	T4		T1	T2	T3	T4	T1	T2	T3	T4
U1	20	8	5	4	1	4	3	2	U6	10	2	3	2	0	0	0	0
U2	5	4	3	3	1	3	3	3	U7	1	1	1	1	0	0	0	0
U3	5	9	3	4	1	4	3	2	U8	1	3	3	1	0	0	0	0
U4	7	3	5	4	1	3	2	2	U9	1	0	0	0	0	0	0	0
U5	5	7	15	4	1	3	2	1	U10	4	3	3	1	0	0	0	0
Avg.	8,4	6,2	6,2	3,8	1	3	3	2		3	2	3	1	0	0	0	0

Table. 7 shows the assessment result in an evaluation matrix of results. For each user we present the result of performing each task. The average completion time of task 1 was reduced from 8.4 s to 4s. For task 2 it was reduced from 6.2s to 2s. The task 3 it was reduced from 6.2s to 3s. Finally, the task 4 was reduced from 3.8 s to 1s. In the current SGA the Task 1 was the one that took the longest time in average to be completed by the users. On the other hand, the maximum optimization was reached on the task 4. The reader must note that the new SGA application mobile distribution avoided the screen glide because the widgets can be found at first sight. Also, this distribution provides a most optimal functional solution for the alphabetic previous distribution. So far so good, we have performed an evaluation presenting preliminary results supporting those benefits claimed by our approach.

8. CONCLUSIONS AND FUTURE WORK

A web information system consists mainly of functional components described in terms of their behaviors and interfaces and the interconnections between components [17]. For this reason, the behavior of the user's navigation can be modeled as a Markov chain, through finite states within a transactional Web system that allow improve the software obtaining an optimized user interface promoting a more intelligent navigation. Nowadays, the user is very familiar with the closeness matrix, for this reason this approach proposes a design centered in them to interrelate software functionality in a model that they understand more easily. This methodology allows prioritizing the content of a legacy application in mobile devices, taking into account the reduced viewport with the flow of information from a user centered-design approach. Firstly, the new distribution allows the users to find faster the widgets they require. Secondly, the redistribution disposes the widgets accordingly their functional affinity. Finally, this new layout decreases the effort required to search a widget in the page, which reduces the time to complete the tasks and the size of the scrollbar. The hypotheses the Markovian model has been theoretically proved.

In addition, there is a future work to improve the development the tool that supports this Markov approach for the design of mobile interfaces, in order to measure its benefits, through a controlled experiment to evaluate its proactivity according to the use of the system to became an evolutive software. Also, the model will analyze the

way in which the efficiency of the operation of a software application can be continuously improved by minimizing the operations performed (i.e. clicks, zoom, and scroll) and their cost (energy/time) according to the dynamic distribution proposals.

REFERENCES

[1]   S. Abram, "Millennials: Deal With Them!," *Texas Libr. J.*, 2006.
[2]   M. Stanley, "The Mobile Internet Report," *Most*, pp. 9–59, 2009.
[3]   T. O'Reilly, "What is web 2.0?: design patterns and business models for the next generation of software. O'Reilly.," no. 65, pp. 17–37, 2005.
[4]   D. Zimmerman and T. Yohon, "Small-screen interface design: Where are we? Where do we go?," in *IEEE International Professional Communication Conference*, 2009.
[5]   H. Hoehle, R. Aljafari, and V. Venkatesh, "Leveraging Microsoft's mobile usability guidelines: Conceptualizing and developing scales for mobile application usability," *Int. J. Hum. Comput. Stud.*, vol. 89, pp. 35–53, 2016.
[6]   J. M. Rivero *et al.*, "Improving Legacy Applications with Client-side Augmentations," in *18th International Conference, ICWE 2018, Caceres, Spain, June 5-8, 2017, Proceedings*, 2018.
[7]   P. Taylor, P. Kortum, M. Sorber, P. Kortum, and M. Sorber, "International Journal of Human-Computer Interaction Measuring the Usability of Mobile Applications for Phones and Tablets Measuring the Usability of Mobile Applications for Phones and Tablets," no. August, 2015.
[8]   J. Brooke, "SUS - A quick and dirty usability scale," *Usability Eval. Ind.*, pp. 4–7, 1996.
[9]   A. Bangor, T. Staff, P. Kortum, J. Miller, and T. Staff, "Determining What Individual SUS Scores Mean : Adding an Adjective Rating Scale," vol. 4, no. 3, pp. 114–123, 2009.
[10]   J. E. Cook and A. L. Wolf, "Discovering models of software processes from event-based data," *ACM Trans. Softw. Eng. Methodol.*, vol. 7, no. 3, pp. 215–249, 1998.
[11]   J. A. Whittaker and J. H. Poore, "Markov analysis of software specifications," *ACM Trans. Softw. Eng. Methodol.*, vol. 2, no. 1, pp. 93–106, 1993.
[12]   C. J. Cheng, S. W. Chiu, C. B. Cheng, and J. Y. Wu, "Customer lifetime value prediction by a Markov chain based data mining model: Application to an auto repair and maintenance company in Taiwan," *Sci. Iran.*, vol. 19, no. 3, pp. 849–855, 2012.
[13]   C. S. D. . Murphy, K. P.(University of California, Berkeley, *Inference and learning in hybrid Bayesian networks.* 1998.
[14]   H. Thimbleby, P. Cairns, and M. Jones, "Usability analysis with Markov models," *ACM Trans. Comput. Interact.*, vol. 8, no. 2, pp. 99–132, 2001.
[15]   J. D. Muther, R., & Wheeler, *Simplified systematic layout planning*. Management and Industrial Research Publications., 1994.
[16]   C.-Y. Mao and Y.-S. Lu, "Testing and evaluation for Web usability based on extended Markov chain model," *Wuhan Univ. J. Nat. Sci.*, vol. 9, no. 5, pp. 687–693, 2004.
[17]   C. Yanchun and W. Xingpeng, "A Security Risk Evaluation Model for Mashup Application," 2009.
[18]   A. Chohan and A. Bibi, "Optimized Software Product Line Architecture and Feature Modeling in Improvement of Spl," in *International Conference on Frontiers of Information Technology*, 2017.
[19]   C. C. Nwobi-Okoye and S. Okiy, "Application of game theory to software user interface evaluation," *Cogent Eng.*, vol. 5, no. 1, pp. 1–18, 2018.
[20]   V. Cajas, M. Urbieta, Y. Rybarczyk, G. Rossi, and C. Guevara, "Portability Approaches for Business Web Applications to Mobile Devices: A Systematic Mapping," Babahoyo, 1, 2018.
[21]   M. Lázaro and E. Marcos, "Research in Software Engineering : Paradigms and Methods," *CAiSE Work.*, pp. 517–522, 2005.
[22]   M. Zelkowitz and D. Wallace, "Experimental Models for Validating Technology," *Computer (Long. Beach. Calif).*, no. May, pp. 23–31, 1998.
[23]   K. H. Ã, "Current practice in measuring usability : Challenges to usability studies and research," vol. 64, pp. 79–102, 2006.
[24]   U. Indoamerica, "Sistema de Gestión Académica," 2017. [Online]. Available: https://sga.uti.edu.ec/login?ret=/. [Accessed: 29-Nov-2018].
[25]   Mobincube, "mobincube," 2018. [Online]. Available: https://www.mobincube.com/es/. [Accessed: 01-Dec-2018].