Sandbox UFPS - cloud development platform for server management, creation and deployment of web applications of academic use

Fredy H. Vera R.[1]*, Boris R. Pérez Gutiérrez [1] and Fernando J. Torres Bermúdez[1]

[1]Universidad Francisco de Paula Santander. Grupo de Investigación y desarrollo de ingeniería del software – GIDIS. Programa de Ingeniería de Sistemas. Cúcuta. Colombia
{fredyhumbertovera, borisperezg}@ufps.edu.co,
fernandojoset@gmail.com

Abstract. This work consists in the implementing of Sandbox UFPS, this is a development platform in the cloud and server management for academic use, which allows the administration, configuration and deployment of web applications, making use of web technologies such as PHP, Python, JSP, .Net (apache, tomcat, glassfish, IIS) and servers of relational and non-relational databases (MySQL, PostgreSQL, MongoDB), to be used by students and teachers of the Systems Engineering program at Universidad Francisco de Paula Santander. The process of development of this platform was carried out, first identifying the needs, tools and services used in development of this type of applications, then implementing, documenting and testing these services obtaining an integrated platform where students can create your projects, within each project the students can deploy different instances of servers and tools as they need; and using version control GIT (GITLab) for a collaborative work. The benefits are significant, students can deploy and host web applications in just a few steps, learn to configure, install their web applications and have a development environment close to what they will find in their working lives. To improve availability, backup and security of the information handled by Sandbox, a load distribution between machines is being designed and implemented to offer efficiency that allow to use the advantages of cloud computing and improve the development platform.

Keywords: Cloud computing, platform as a service, scrum, server management, web applications.

1 Introduction

The National Institute of Standards and Technology (NIST) US defines cloud computing as a model that allows access to the network, to request and a convenient way, to a shared group of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be implemented quickly with minimal management effort or interaction of the service provider [1]. The definition formulated by

pp. 65–75; rec. 2015-08-22; acc. 2015-09-21

George Reese is [2]: "The cloud is not just a fancy way of describing the Internet. While the Internet is a key foundation for the cloud, the cloud is more than just the Internet. The cloud is where you go to use technology when you need it, as long as you need, and not a minute more. You do not install anything locally, and do not pay for the technology when you are not using it. "In this way, the Cloud becomes a service that we can consume under our own needs.

The services offered may consist of hardware or software, or a combination of both, and is conceptually in the Cloud online. Another factor contributing to the concept of the Cloud by its intangible essence is the fact that neither the user nor the developers know where or how your service is hosted (and certainly do not need to know). Or perhaps even the provider of cloud services could determine immediately where is an instance of the services within its data center [3]. Joyanes in [4] sets out in clear terms the Cloud Computing model makes a description of their architectures and outstanding models. It emphasizes that the cloud is the technological platform per excellence of this decade and the future of computing.

A study by Gartner [5], pose a magic quadrant where are detailed the infrastructure providers in the cloud leaders today (2014). It can be seen that leaders are Microsoft with Azure and Amazon Web Services being the last one the most important in the market; Google appears in the visionaries' quadrant and evolving away from Azure and Amazon. Then the major computing platforms are defined in the cloud.

Amazon Web Services: Provides a wide range of computer services, storage, databases, data analysis and big data, applications and implementations that help organizations to move quickly, reduce IT costs and scale applications [6].

Google Apps: is a set of messaging and collaboration software for businesses. The main services are offered in Google Apps are: email (Gmail), instant messaging (Google Talk), calendar (Google Calendar), document management (Google Docs) and site management (Google Sites). Google also offers REST API services which are called the Google Data API for programming not only for Google Apps, but also for other web services offered by Google as Google Maps, YouTube and Google Health. It focuses on software as services (SaaS).

Google App Engine: Corresponds to infrastructure (IaaS) and platform as a service (PaaS) of Google. It is an environment for developing and deploying web applications on Google's infrastructure. The App Engine supports Python and Java as the main programming languages, automatic scaling and load balancing. It also provides a data warehouse that allows to create, retrieve and delete functions and data, provides Google Data APIs to access to the components of Google Apps such as mail, calendar, search and Docs.

Microsoft Azure: A computer and Internet services platform hosted in data centers managed or supported by Microsoft. It includes many independent features, with relevant services to developers that can be used separately or together [7]. With Azure is possible to build, deploy and manage applications quickly. They can be integrated the applications in the cloud with existing traditional IT environment. Azure has 4 main models for building and running applications: 1) Virtual machines, 2) Cloud Services, 3) Websites and 4) Mobile services.

The references are few who are in academic applications of cloud computing, in Colombia EAFIT University has a datacenter which offer development environments to students, access to virtual machines and different services and infrastructure and platform tools as service [8]. Universidad de los Andes - Colombia, in its lab Computer and Systems Engineering, has a server system for managing computer projects oriented to students' thesis or computer projects, offer virtual machines, access to databases, storage and specialized software [9].

Knowing the definition of cloud computing, services offered and major suppliers, can now contextualize the problem that is showing in the development of projects semester and graduation projects of students in Systems Engineering at The Francisco de Paula Santander University, in these works students implement computer solutions that solve problems for companies in the region. These works are essential because these developments allow students to learn to conceptualize the theoretical with the praxis, thereby making more complete and competitive learning. They have been presented loss of information (source documents and model codes), delay in deploying applications, collaborative work difficulty, and difficulty in getting the servers to deploy applications. Hence the idea of developing this platform arises. The concept of Sandbox responds to a separate test environment from the production environment, taking this definition to this context we can say that Sandbox is a development platform in the cloud where students find development tools and deployment of web applications.

Sandbox was created as an alternative for students where they can develop, deploy, test and manage their applications without having to rely on payment of a hosting or servers shifts at the University or rely on the availability of administrators. This idea was born in the Systems Department of the university in the second half of 2011 using technologies such as PHP and JSP with relational databases PostgreSQL and MySQL, a year later students of the subject Web Applications make as project of it Tools that would optimize the platform, is how in the second half of 2012 these tools are integrated into the platform and give more strength to the project. Due to the success of these implementations the need to give more coverage to the tool was seen, by which is chosen to take a model of successful cloud, and it is to simulate the implementation by Amazon Web Service (AWS) it is thus a platform for the administration of development on a cloud environment is created, which is available anytime and anywhere, allowing to develop, configure and deploy applications using different tools and services, saving time and money to students. This development can be replicated and used as reference for software development companies both nationally and internationally.

In this article, section 2 shows the research work methodology, then Section 3 presents the results and functionalities are specified and services offered by the platform UFPS Sandbox, finally the findings and the profits are set.

2 Methodology

The research carried on this project is the Applied Technological Research, Padrón [10] states that: 1) it starts on a problematic situation that needs to be tapped and improved. The situation is systematically described under relevant criteria, it starts from some theories that help solve the problem, and a prototype which is verified and validated so that solves the problem situation is proposed, finally give conclusions and recommendations. In this case the problem situation corresponds to the development, deployment and configuration of web applications from Systems Engineering students, which is important to improve, control and manage, the theories that will be used to improve the situation are: agile development methodologies, cloud computing, software engineering and web programming. Based on these theories the procedure is to analyze, design and implement development platform Sandbox. Finally, an evaluation and validation of the platform is made to test it, refine it, and correct any shortcomings and ensure that solves the problem situation proposed.

It is based on the statements made by Padron and the principles of agile software development, the next step is to establish the methodological design of this research, which is summarized in Figure 1. The methodology of the agile development that is fundamental basis for this work is Scrum [11], which allows developing the system through iterations, in each iteration a part of the software product is obtained, which is validated by the customer, to ensure it accomplishes with what is expected. For the project the following roles scrum were defined: 1) the scrum master was the project manager, 2) development team was formed by its authors, 3) the product owner corresponds to: a teacher and a group of students from Systems Department UFPS.

Proposed by Scrum follows cycle is adapted: the research phase of the project includes planning of scrum and the selection of detailed requirements using use case model; the design phase includes iteration planning and defining the architecture of the iteration; the development phase corresponds to the execution of the iteration and where a product is built increased and finally the validation phase contains demonstration and retrospective of the iteration.

The performed iterations were: 1) Overview of the Sandbox platform, 2) Functionalities of student. 3) Functionalities of Administrator. 4) Functionalities of teachers. 5) Functions for monitoring and control platform failures. Use cases to specify the requirements defined in each sprint implement were used. The following summarizes each of the phases of the methodology carried out.

2.1 Research

At this stage will be carried out an analysis of services, tools and features that will be included in the Sandbox platform. They way to integrated it into the platform was studied. System requirements were determined, the problem analysis was executed, and the favorable technological framework environment for system development of each iteration was generated.

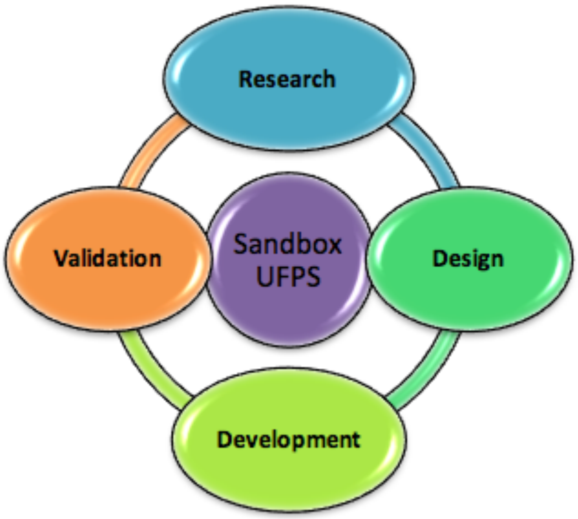

Fig. 1. Methodology

2.2 Design

It was executed in a set with the previous stage, the scope and limitations of each iteration was defined, risks were identified, the budget and timetable of development were adjusted, the design of the system architecture was made, which will be expanding and improving with each iteration.

2.3 Development

At this stage, the source code that fulfill the proposed design and the requirements defined for iteration was created, the obtained prototype was tested and evaluations required were made for the robustness of the system. The adjustments to the design of the system architecture were performed. Daily follow-up, control and synchronized meetings are held between members of the development team and a weekly with the product owner and scrum master.

2.4 Validation

Given the functional prototype, the instruments to assess and validate their performance by students and teachers in order to receive an important feedback to the system were defined, obtaining at the end a closer product to the needs of users. The adjustments that are required were carried out, and the lessons learned, findings and the relevant aspects of the iteration were registered.

3 Cloud Platform Sandbox UFPS and Results

As the main result was obtained a platform in the cloud that provides services and tools for development of web projects by students and teachers of the Engineering Systems program UFPS. With this platform, students can access to a development environment close to what they will have in working life where they can configure, manage, and deploy web applications at any time, from anywhere and at no cost.

On the platform students can create their development projects, creating instances in database managers, and various web technologies such as JSP, PHP, .NET and Python.

In the past 6 semesters the average of applications that are hosted in Sandbox are 71 per semester, this number is obtained from the statistics compiled by the administrators of this tool in the past three years. See Table 1.

	2nd Semester 2012	*1st Semester 2013*	*2nd Semester 2013*	*1st Semester 2014*	*2nd Semester 2014*	*1st Semester 2015*
Number of Applications per Semester	42	63	66	72	84	96

Table 1. number the applications used Sandbox.

3.1 System Architecture

The system architecture consists of a server hardware of the following characteristics: A QuadCore 2.4GHz, with 8GB of RAM and disk space of 250 GB, this hardware is distributed in virtual machines each with one core processor, 2GB of RAM and 50 GB of disk space. Therefore the Sandbox system in a complex of 3 VMs and whose architecture can apre-astern in Figure 2, in this same way you can see how a balanced load between servers is done. Which is done leaving all data-to Processing in the Sandbox 2 and can be seen in the following figure:

Sandbox server 1: Initial server where the application of these tools administration is saved, administrative application built JSP on a database administrator in PostgreSQL that manage the database; on this server codes are also housed in different technologies such as Python, PHP and JSP system users. The server is responsible for performing all the processing code, freeing the data processing managers it gives database.

Sandbox Server 2: Server where the database system users and the repository projects are housed. In this server only consultative processes and data storage is done.

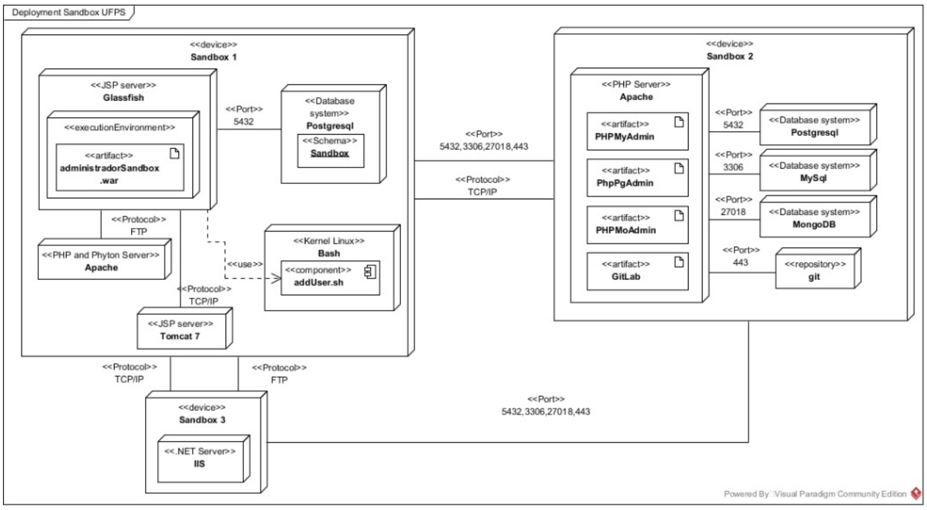

Fig. 2. Sandbox UFPS Architecture

Sandbox Server 3: .Net server where deployments projects on .Net users were made. In this sandbox server as in 1 only code processing is performed.

The idea of this architecture is to distribute the server load by locating databases on a different machine where project operations are conducted; this way the server load times of the requests made are lightened.

3.2 Functionalities of UFPS Sandbox.

To detail the functionalities provided Sandbox UFPS the use case diagram UML was used. First, actors for this system are defined, which are detailed below:

- Student: It is the main user of the application is responsible for creating the projects and define instances of the technologies to be used.
- Teacher: Responsible for the review of projects can also create instances of the technologies sandbox provides.
- Administrator: Actor who is responsible for the configuration and system support. Create and assign permissions and access levels for students and teachers.

These roles and share some features and have also functionalities unique to each as seen below:

3.2.1 New Project: functionality that allows the user to register a project on the tool, this process begins selecting instances needed for the project, then if necessary add users to the project and finally is saved, in this step the Sandbox system generates the necessary deployment of instances on the server settings, then performs the association of these configurations to other accounts related to the project users, these users can be students or teachers.

This functionality will only be executed by the student and teacher users.

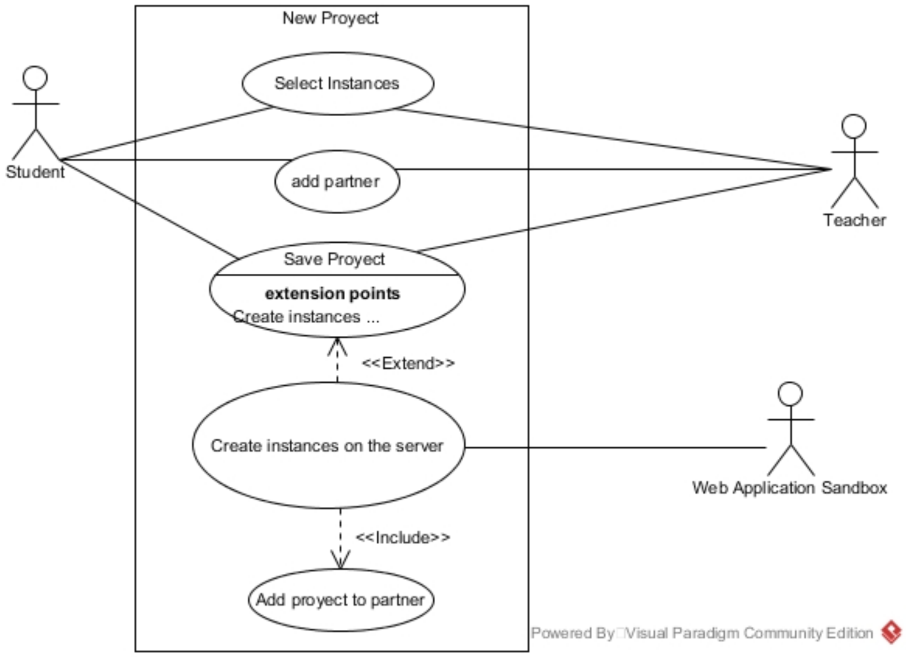

Fig. 3. New Project Use Case

3.2.2 Deployment Instances: This functionality is used by the user to upload their projects to the server, so the cost of two types of deployment for instance, the first is upload applications through the application interface and second one is increasing their databases through each handler.

In the first type of deployment tool uses two types of settings, which are through the server scripts run custom commands to deploy in the server, the other is through file transfer upload scripts to the server. These processes are executed by the tool automatically; in the second type interfaces handlers databases are used, such interfaces are: PhpMyAdmin, PhpPgAdmin and PhpMoAdmin

This functionality is available for student and teacher users.

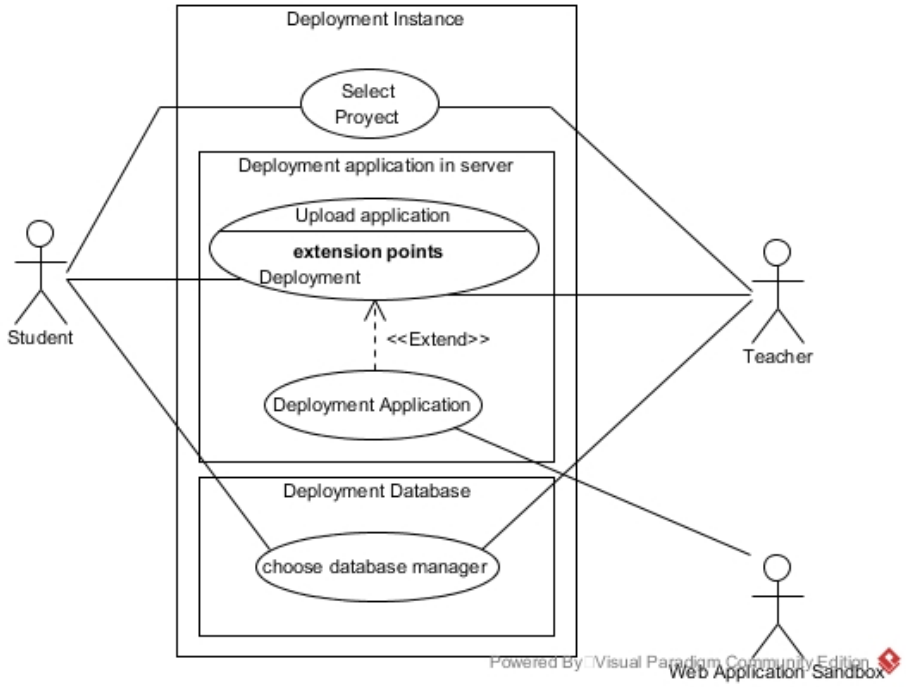

Fig. 4. Deploy Instance Use Case

3.2.3 Add Instance: Functionality used to allow the user to configure how to use technologies added to the server, choosing between using a script on the server for deploying or using the FTP in a file server.

This feature is only available to the administrator.

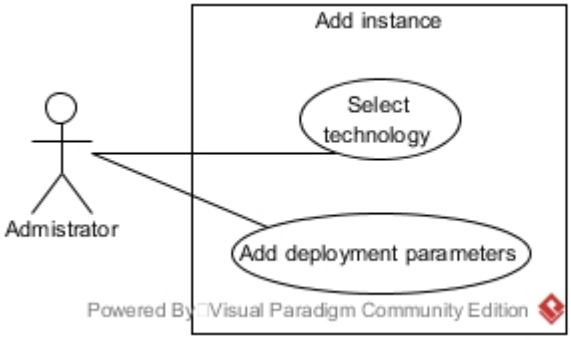
Add instance
Select technology
Add deployment parameters
Admistrator
Powered By Visual Paradigm Community Edition

Fig. 5. Add Instance Use Case.

3.2.4 List Projects: functionality that allows users to list the projects by range permitted and thus select the project that want to see and access to this; the use case starts when the application displays a list of projects by type of user, the user selects the project view and then can access the instance which want to check, so the Sandbox runs the deployment for instance. This functionality is available to all users with a slight variation of permits, which are:

Student: Can only access the projects that they have been added or created.
Teacher: Can access the projects that have been added to their classes.
Administrator: Can view all projects.

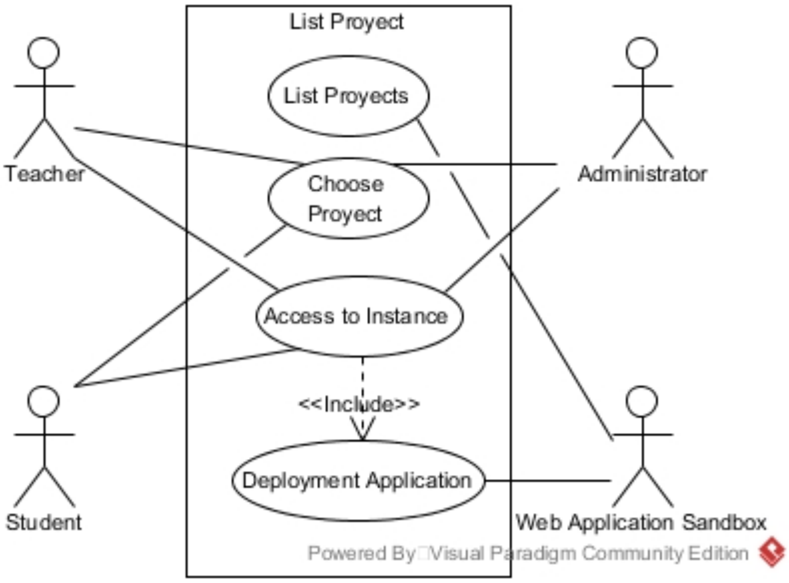

Fig. 6. List Projects Use Case

4 Conclusions

The Sandbox is a software tool created to bring the student a little closer to the real world of software development, so it was necessary to choose an instances structure that are used on a project, these instances are a variant of technologies needed to make the product as close to a real-world solution in this type.

This makes easy the management of projects both by students and teachers as well as the integration of the development team for such projects; it helped to these tools to have a high degree of acceptance in the academic community of the Engineering Systems UFPS curriculum.

On the other hand, was created a tool that allows to engage a wide number of technologies thus ensuring a larger life cycle, this solution was suggested considering that previous versions got outdated in a period of time of 2 or 3 years.

The Sandbox is a software tool created to bring the student a little closer to the real world of software development, so a structure of instances that are used on a project were chosen, these instances are a variant of necessary technologies so that the product is as close to a real-world solution in this type.

This facilitated the management of projects for both by students and teachers as well as the integration of the same development team for such projects; this helped these tools have a high degree of acceptance in the academic community of the curriculum of the Engineering Systems UFPS.

References

1. C. IDG, «Libro blanco "Hablando Cloud",» 2012. [En línea]. Available: http://www.idg.es/hablando_cloud/Default.aspx. [Último acceso: 20 03 2013].
2. G. Reese, Cloud application architectures, O'Reilly Media, 2009.
3. J. Mestas, «La nube como plataforma computacional,» 9 8 2010. [En línea]. Available: http://geekswithblogs.net/gotchas/archive/2010/08/09/la-nube-como-plataforma-computacional.aspx. [Último acceso: 2014 7 31].
4. L. Joyanes Aguilar, Computación en la Nube: Estrategias De Cloud Computing En Las Empresas, Alfaomega, 2012.
5. Gartner, «Magic Quadrant for Cloud Infrastructure as a Service,» 28 5 2014. [En línea]. Available: http://www.gartner.com/technology/reprints.do?id=1-1UKQQA6&ct=140528&st=sb. [Último acceso: 13 8 2014].
6. Amazon Web Services, «Productos y servicios,» 2014. [En línea]. Available: http://aws.amazon.com/es/products/. [Último acceso: 13 8 2014].
7. Microsoft, «¿Qué es Azure?,» 4 2014. [En línea]. Available: http://msdn.microsoft.com/es-es/library/azure/dd163896.aspx. [Último acceso: 5 8 2014].
8. «Universidad EAFIT historia de exito,» 2011. [En línea]. Available: http://www.greenews.com.mx/pdf/centro-de-datos/casos-de-exito/Caso%20Estiudio%20-%20UNIV%20EAFIT%20%20-%20%20(SA-NCCS29-LA)%20-pdf.pdf. [Último acceso: 24 9 2015].
9. Universidad de los Andes - Colombia, «Laboratorio de Sistemas,» 2015. [En línea]. Available: https://labsis.uniandes.edu.co/index.php/servicios-estudiantes. [Último acceso: 24 9 2015].
10. J. Padrón G, «Bases del concepto de "investigación aplicada",» 2006. [En línea]. Available: http://padron.entretemas.com/InvAplicada/index.htm. [Último acceso: 21 3 2013].
11. Proyectos agiles ORG, «Proyectos agiles.org,» 2014. [En línea]. Available: http://www.proyectosagiles.org/. [Último acceso: 10 7 2014].