SADIO Electronic Journal of Informatics and Operations Research

http://www.sadio.org.ar

vol. 10, no. 1, pp. 78-91 (2011)

JOnE: JEE Online Development Environment "Programming on the cloud"

Pablo Larrimbe[1] Pablo Salgado[2] Vanesa Maiorana[3]

[1] Facultad de Ingeniería
Universidad Argentina de la Empresa
Lima 717. C1073AAO
Ciudad Autónoma de Buenos Aires
Buenos Aires, Argentina
e-mail: plarrimbe@gmail.com
Tel: 0800-122-UADE (8233) . 4000-7600

[2] Facultad de Ingeniería
Universidad Argentina de la Empresa
Lima 717. C1073AAO
Ciudad Autónoma de Buenos Aires
Buenos Aires, Argentina
e-mail: pdsalgado@gmail.com
Tel: 0800-122-UADE (8233) . 4000-7600

[3] Facultad de Ingeniería
Universidad Argentina de la Empresa
Lima 717. C1073AAO
Ciudad Autónoma de Buenos Aires
Buenos Aires, Argentina
e-mail: vanesamaiorana@gmail.com
Tel: 0800-122-UADE (8233) . 4000-7600

Abstract

We present in this paper a new proposal for the development of software applications, called JOnE (JEE Online Environment). JOnE is a web environment for Java programming. Through the use of the proposed idea and the tool developed to support it, it is possible to develop a complete Java application, without having a computer with the development environment installed and configured on it. With an internet conection and a web browser, the development team could build the software system and modify it collaboratively from anywhere.

The main goal of this work has been achieved: to build a novel tool in the area of programming environments. The tool is not a very complex one and the prototype we developed has not an extensive list of functionalities, but it presents a change in the programming approaches. In this paper we present the new development approach we propose and we demonstrate it is possible through the prototype of the developed tool. We also include the description about its functionalities, its benefits and limitations.

Keywords: : Development environments, Web IDE, Programming on the cloud

1 Goals

The main goal of this work is to make possible that the programming process can be done by every user, independently of the environment and tools he has already installed in their computers. To achieve that goal, we present a *web environment* that integrates all the necessary tools for the development of software applications. The user can concentrate in the development itself, without installing and configuring the development environment, which generally needs extra efforts and brings configuration and infrastructure problems.

The web Server, the data base and the IDE to use, are installed in the Server side. The user only concentrates in the programming. Once the user has programmed the software application, the proponed environment makes the project compiling, the web Server and data base administration, so this tasks are transparent and friendly to the final user (the programmer). This is possible, although we have to take into account certain limitations related to web technology compared to desktop programming environments. It is not a goal in this paper to focus on that differences.

To complete the definition of the main goal of this work, we describe the steps that have to be followed by a typical user to create a basic web application are. First, he has to create a new user account or login to the systems if he has one. Then he can create a new project or select an existing one. After that he can create project files, which can be *Java Script, JSP, Css, Java or Servlet.* The user can also organize his files thorug folders, creating the structure of the application. The user can compile the project any time he wants and the system will show the found errors during compilation. Once the project has been finished, the user can deploy, download the .war file and test the application.

2 JOnE: Evironment for programming on the cloud

The motivation to build JOnE has arrived from the needs of a development team to build a web page in Java. To build such a page, each member of the team needs to install an ide such as Eclipse (mostly, ides are desktop applications). They will need also a database and a web server such as Tomcat. And of course they will need a computer with certain characteristics and conditions depending on the requirements.

JOnE is an Iintegrated Development Environment (IDE) that functions in the web. The ide allows doing all the tasks involved in a project of software development: programming, data persistence (in a relational database), testing and deployment. The user can create, modify and delete Java web projects.

A Java web project is a set of files of different types: Java Script, css, Servlets, .jsp, .java, .jar (libraries), and a descriptor file of the web application (.xml). The user can edit the mentioned files and do the upload of the libraries he needs to use.

3 State of the art

Cloud computing (computing on the internet) is a novel paradigm that allows to store information on Internet servers without installing the applications in the local workstations of clients. This paradigm is based on the “software as service” modality. There are many examples of cloud computing applications: Amazon EC2, Google Apps, eyeOS and Microsoft Azure. Those projects provide common applications online, accessible from a web explorer. The software and the data are hosted on a server.

Cloud computing applications are very varied and offer many types of services to clients. However, in the area of development environments, there are many things to do. Nowadays, there are not many tools or applications such as the proposed in this paper. Some of them can be categorized as text editors online. Examples of them are Bespin, a beta project by Mozilla Labs, and Ecoder, another beta project from Gmeditor.

Both can be tested free from their website. They are both in beta versions, which means that they are still in a process of improving and maturity has not been achieved yet in the area of software development industry on the web.

There is another application most similar to a development environment, that is also on beta version. This is called WebDebStudio and was developed to code files and build ANSI c and c++ projects.

4 JOnE Functionalities

In this section, we describe the functionalities that are included in the JOnE developed prototype:

4.1 Login

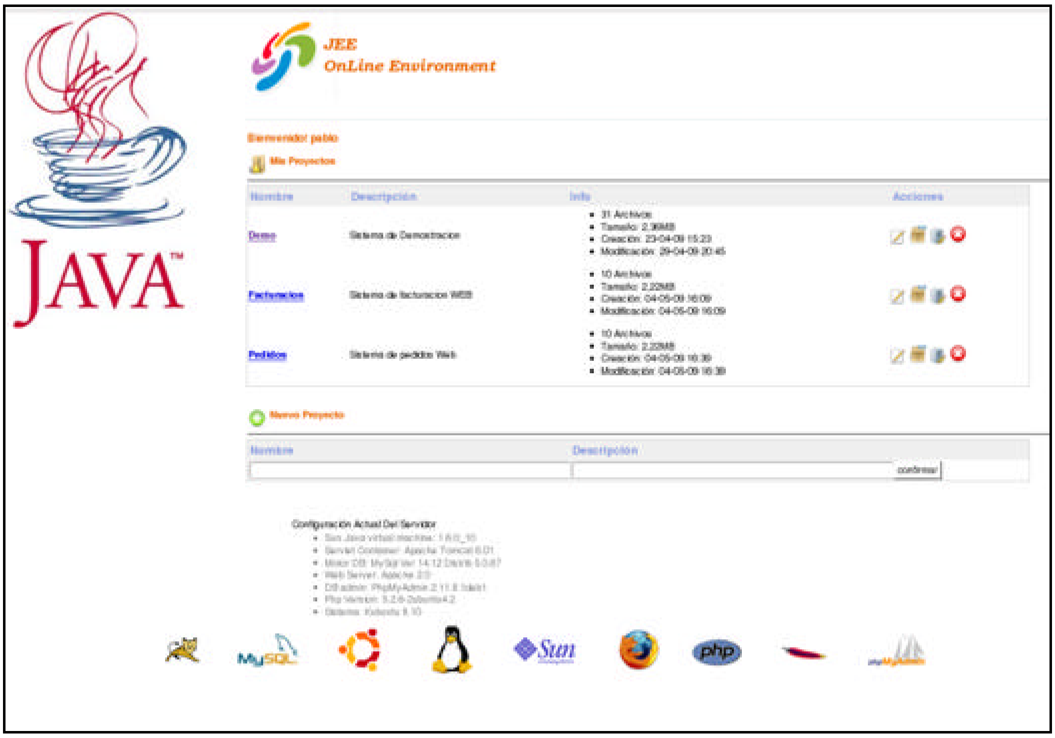
The user can access to the system by entering user and password. When he enters, he can work on the projects already created by him or he can build new projects.

4.2 Project administration

The user can add, edit or delete his own files that are included in his own projects. Besides, he can download .war files, by downloading the standard file of a JEE web application. This file can be installed later in every web server which applies to the JEE web application standard, such as Jetty or Tomcat.

Through the project administrator module, the user can administrate his tables using the table administrator of the PhpMyAdmin database, which is integrated to the JEE Online Environment (JOnE). (See figure 1).

The user can access directly to his databases using the permissions required. For each application, there is a database associated whose name is equal to the application name.



Bienvenido! pablo

Mis Proyectos

Nombre	Descripción	Info	Acciones
Demo	Sistema de Demostracion	31 Archivos Tamaño: 2.36MB Creación: 23-04-09 15:23 Modificación: 29-04-09 20:45	
Facturacion	Sistema de facturacion WEB	10 Archivos Tamaño: 2.22MB Creación: 04-05-09 16:09 Modificación: 04-05-09 16:09	
Pedidos	Sistema de pedidos Web	10 Archivos Tamaño: 2.22MB Creación: 04-05-09 16:39 Modificación: 04-05-09 16:39	

Nuevo Proyecto

Nombre	Descripción	
		confirmar

Configuración Actual Del Servidor

- Sun Java virtual machine: 1.6.0_10
- Servlet Container: Apache Tomcat 6.0.1
- Motor DB: MySql Ver 14.12 Distrib 5.0.67
- Web Server: Apache 2.0
- DB admin: PhpMyAdmin 2.11.8.1deb1
- Php Version: 5.2.6-2ubuntu4.2
- Sistema: Kubuntu 8.10

Figure 1. JOnE. Project Administration Screen

4.3 Database administration

JOnE is integrated with PhpMyAdmin (PMA) for the database administration using MySQL. This integration allows the change of web Servers in a transparent way for the user. It also allows the user unified login.

Through this functionality, the user can create, edit and delete tables, he can define simple queries (SELECT queries) of more complex queries (using JOIN, UNION sentencies). See Figure 2.

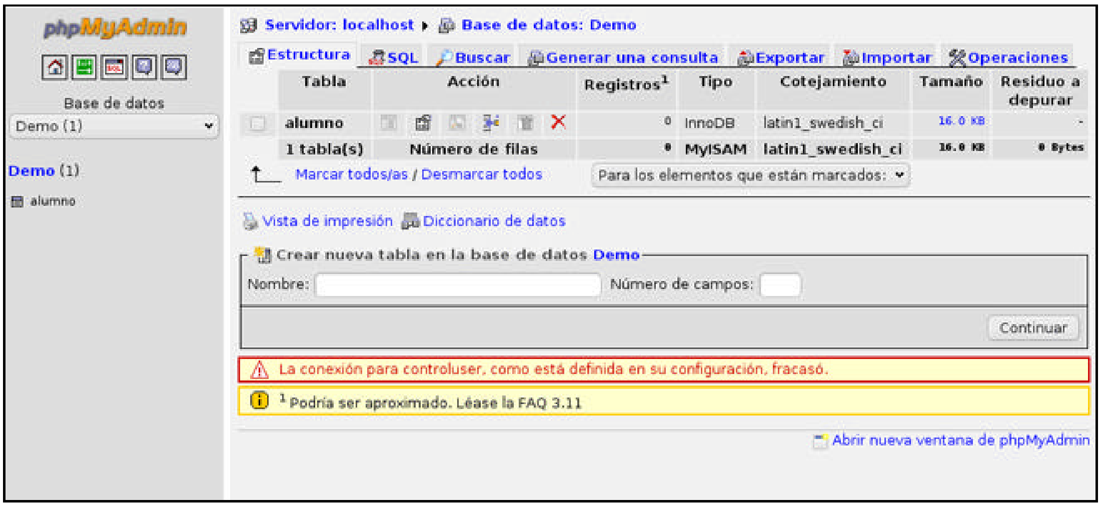


	Tabla	Acción	Registros[1]	Tipo	Cotejamiento	Tamaño	Residuo a depurar
□	alumno	✕	0	InnoDB	latin1_swedish_ci	16.0 KB	-
	1 tabla(s)	Número de filas	0	MyISAM	latin1_swedish_ci	16.0 KB	0 Bytes

Marcar todos/as / Desmarcar todos   Para los elementos que están marcados:

Vista de impresión   Diccionario de datos

Crear nueva tabla en la base de datos Demo

Nombre: Número de campos:

Continuar

La conexión para controluser, como está definida en su configuración, fracasó.

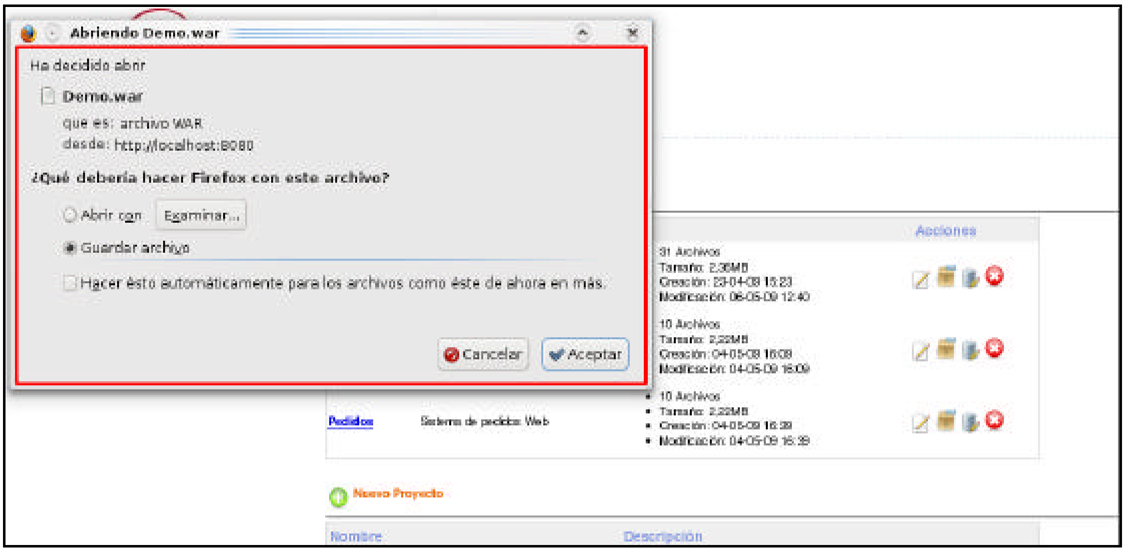
[1] Podría ser aproximado. Léase la FAQ 3.11

Abrir nueva ventana de phpMyAdmin

Figure 2. JOnE. Database administration Screen

4.4 Download war file.

This functionality allows to the user to download the .war file and use it later in his own server. He can also distribute or modify it. (See Figure 3)



Abriendo Demo.war

Ha decidido abrir

Demo.war

que es: archivo WAR

desde: http://localhost:8080

¿Qué debería hacer Firefox con este archivo?

○ Abrir con Examinar...

● Guardar archivo

□ Hacer ésto automáticamente para los archivos como éste de ahora en más.

31 Archivos
Tamaño: 2,36MB
Creación: 23-04-08 15:23
Modificación: 06-05-09 12:40

10 Archivos
Tamaño: 2,22MB
Creación: 04-05-08 16:08
Modificación: 04-05-09 16:09

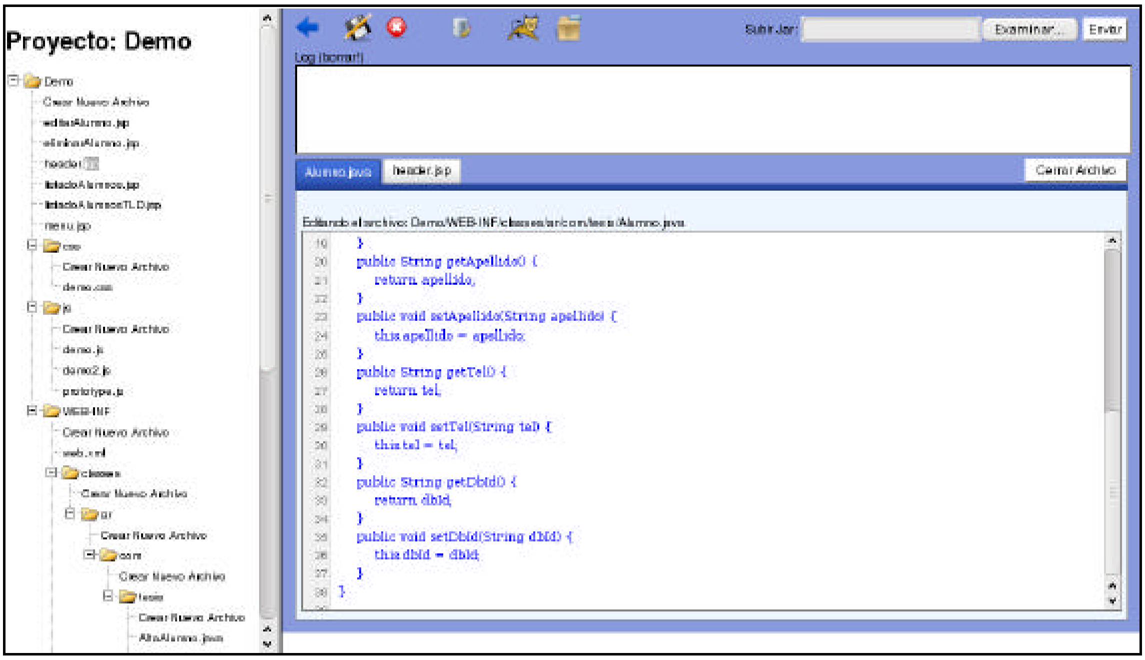
Pedidos Sistema de pedidos Web
- 10 Archivos
- Tamaño: 2,22MB
- Creación: 04-05-09 16:39
- Modificación: 04-05-09 16:39

Figure 3. JOnE. Download of .war file.

4.5 Code editor and integrated compiler.

JOnE has a text editor for the edition of the application files. The user can see many files at a time through the use of tabs. In the case of .java and .jsp files, JOnE saves the source code and makes a compilation of them. In the log file, the user can see the result of the compilation: if it was a successful compilation or the errors obtained.

The integrated compilation is a useful functionality, because the user does not have to wait until the deployment of the application in Tomcat server and execution to see the errors. He can see them while he is programming, because he can do the debugging. The text editor indicates at every moment that the file is being edited. It has the line numbers to facilitate the finding of the compilation errors. (See Figure 4)



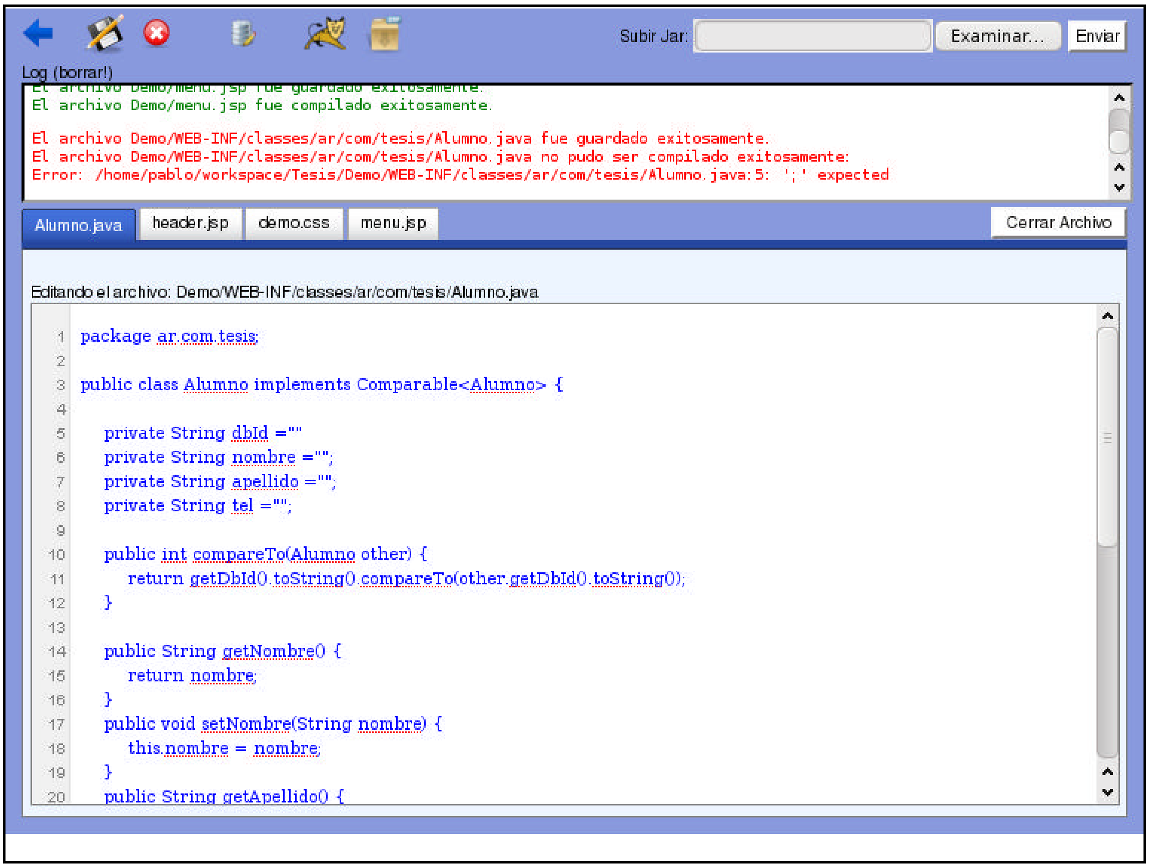
Editando el archivo: Demo/WEB-INF/classes/ar/com/tesis/Alumno.java

```
    }
    public String getApellido() {
        return apellido;
    }
    public void setApellido(String apellido) {
        this.apellido = apellido;
    }
    public String getTel() {
        return tel;
    }
    public void setTel(String tel) {
        this.tel = tel;
    }
    public String getDbId() {
        return dbId;
    }
    public void setDbId(String dbId) {
        this.dbId = dbId;
    }
}
```

Fig. 4. JOnE: Code editor

4.6 Application deployment

This functionality includes the deployment of the built application. JOnE does the deployment in a Tomcat Server, registering in the log file the proble ms encountered. (See Figure 5)



```
El archivo Demo/menu.jsp fue guardado exitosamente.
El archivo Demo/menu.jsp fue compilado exitosamente.

El archivo Demo/WEB-INF/classes/ar/com/tesis/Alumno.java fue guardado exitosamente.
El archivo Demo/WEB-INF/classes/ar/com/tesis/Alumno.java no pudo ser compilado exitosamente:
Error: /home/pablo/workspace/Tesis/Demo/WEB-INF/classes/ar/com/tesis/Alumno.java:5: ';' expected
```

Editando el archivo: Demo/WEB-INF/classes/ar/com/tesis/Alumno.java

```
 1  package ar.com.tesis;
 2
 3  public class Alumno implements Comparable<Alumno> {
 4
 5      private String dbId =""
 6      private String nombre ="";
 7      private String apellido ="";
 8      private String tel ="";
 9
10      public int compareTo(Alumno other) {
11          return getDbId().toString().compareTo(other.getDbId().toString());
12      }
13
14      public String getNombre() {
15          return nombre;
16      }
17      public void setNombre(String nombre) {
18          this.nombre = nombre;
19      }
20      public String getApellido() {
```

Fig. 5. JOnE: Visualization of the log file

When working with .java files, the user can chose to save them as Servlets. In this case, the system updates the web descriptor file .xml automatically, adding to the the directives of the standard. The web descriptor file .xml is editable for the user (as any other file in his application) so that he can customize the directives.

The log file indicates if the files could be loaded and compiled successfully or shows the compilation errors obtained (for compiled files). In the case of Servlet files, the log file contains the update of the web descriptor file .xml. It also indicates if the file was deleted correctly and if the application could be deployed successfully. In that case, it shows the link to test the application.

The log has two different font styles. The actions that could be done, appear in green color. The ones that could no be done, appear in red.

4.7 Upload of .jar file

This functionality allows the user to import external libraries to the web project. In case of importing external files, they will be placed in a folder called WEB-INF/lib of the project, following the standard directives of any JEE web application.

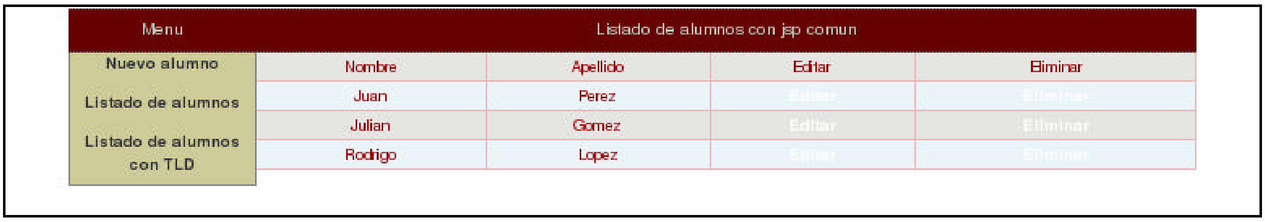
Some libraries are included in the application, such as JSLT support (JSP Standard Tag Library) or the JDBC database connector from MySQL, wich is compatible with the Server version already installed.

4.8 Delete application

Applications can be deleted completely, including the database associated with it and the physical files in the file system.

5. Example of an application built using JOnE

Figure 6 shows a sample application created fully using JOnE as the development environment on the web. The sample application is a system to register students. In this figure, we can see the list of students. The options provided to the final user are: *add*, *edit* or *delete* a student, and *list* students. The list option was developed using Taglibs Tld. Taglibs are an API for JSP in which special variables can be placed, to be filled dynamically at runtime with real data. In this example, the variables are the data of the students.



Menu	Listado de alumnos con jsp comun			
Nuevo alumno	Nombre	Apellido	Editar	Eliminar
Listado de alumnos	Juan	Perez	Editar	Eliminar
Listado de alumnos con TLD	Julian	Gomez	Editar	Eliminar
	Rodrigo	Lopez	Editar	Eliminar

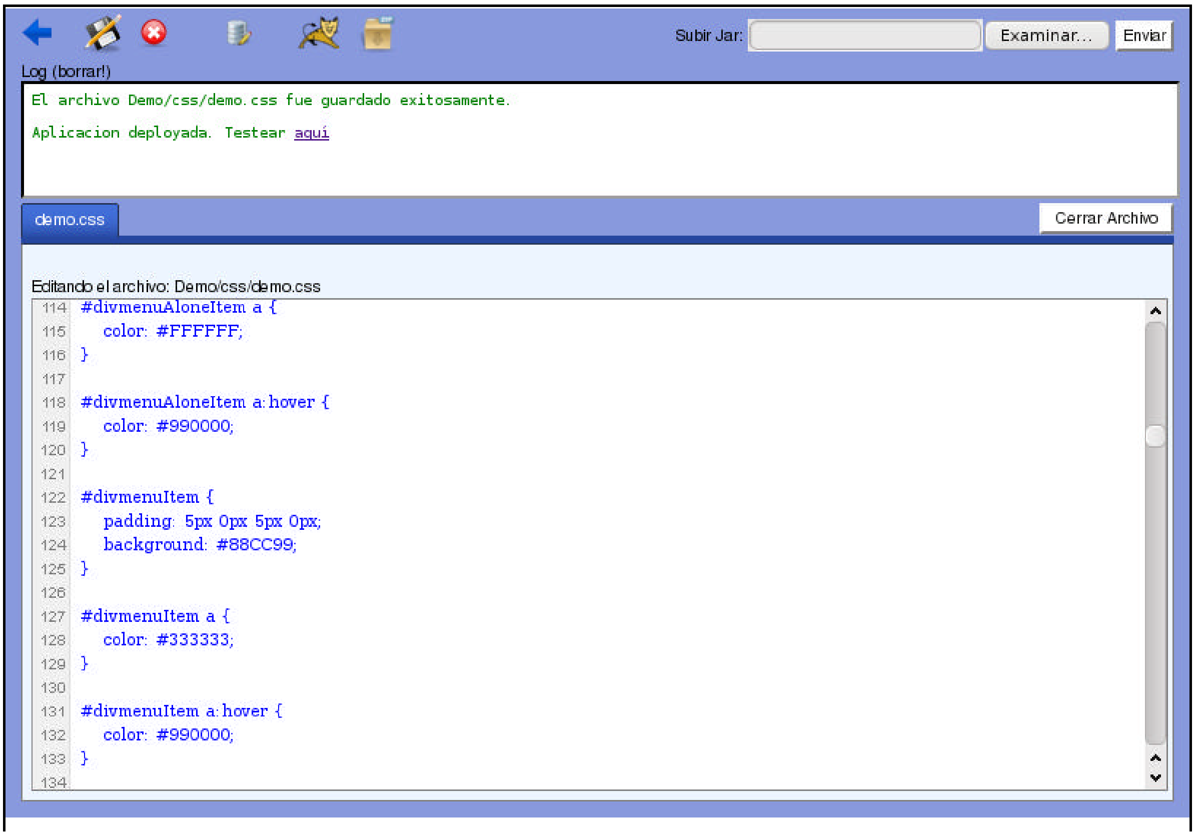
Fig. 6. JOnE: Application built using JOnE

In this section we will show a real scenario to see the benefits of using JOnE. Suppose that we want to change the user interface (look and feel) of the student administration system, built using JOnE. For example, suppose that the user interface was built by a css template by a graphical designer. The designer needs to install an application that allows to modify the template. Without the use of JOnE, he will also need the libraries to be used, install the database of the application and the MySQL Server, install Java, and the Tomcat server to make the necessary updates and test the application.

If he used JOnE, the designer can access directly to the environment, through an internet browser. He modifies the .css, .html, .js or .jsp file, and then he can test the application.

Suppose that we want to change the menu color. The designer needs to do the following:

1. login to the application by entering name and password.
2. Open the necessary files. In this case, he has to open the file “demo.css” and modify the corresponding style.
3. To ensure that the styles are correctly modified, he has to make the application deployment and type the url indicated by the log file generated by JOnE.
4. Verify that the styles are correct and the menu looks like wanted.



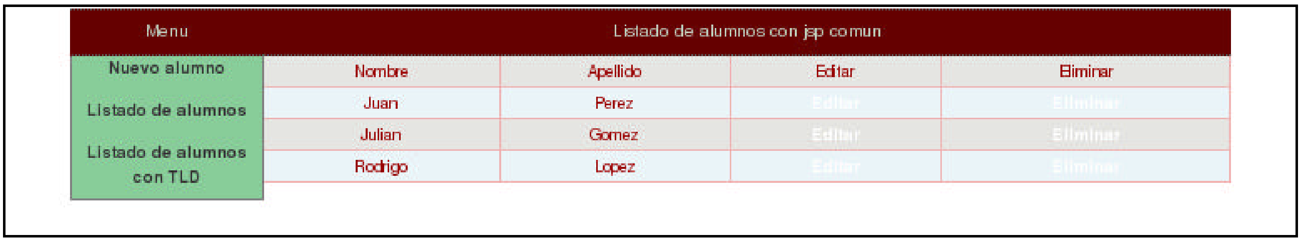
```
El archivo Demo/css/demo.css fue guardado exitosamente.

Aplicacion deployada. Testear aquí
```

Editando el archivo: Demo/css/demo.css

```
#divmenuAloneItem a {
   color: #FFFFFF;
}

#divmenuAloneItem a:hover {
   color: #990000;
}

#divmenuItem {
   padding: 5px 0px 5px 0px;
   background: #88CC99;
}

#divmenuItem a {
   color: #333333;
}

#divmenuItem a:hover {
   color: #990000;
}

```

Fig. 7. Template edition in the example application.

Figure 7 represents the steps 2 and 3 mentioned before. The designer edits styles (demo.css) of the application. In this case he edits the background color of the menu items. Then, he does the deployment of the application. In this example, it is possible to see another feature of the log file. In case that the action executed is the deployment, the user can click inside the log file to open a new tab in the browser or a new browser window (depending on the browser being used). By clicking, the application opens in this tab or window and can be tested. This improved navigation, because the user does not have to abandon JOnE in order to test the application in which he is working.


Menu

Nuevo alumno

Listado de alumnos

Listado de alumnos con TLD

Listado de alumnos con jsp comun

Nombre	Apellido	Editar	Eliminar
Juan	Perez	Editar	Eliminar
Julian	Gomez	Editar	Eliminar
Rodrigo	Lopez	Editar	Eliminar

Fig. 8. Test of the example application, after the changes .

Figure 8 represents the step number 4 mentioned before. In this step, the designer verifies that the new style was applied and the menu color has changed successfully.

In this section, we only showed an example of the use of JOnE. In this particular example, we could see the way a user can be benefited by using JOnE, without the need to install all the necessary tools for the development, or the programming environment. There are many benefits, taking into account that there are many possible scenarios in the update or extension of an application.

If we consider large projects, the development team will be composed by many users with different profiles and functions. Using JOnE, each user can access and modify only what he needs. He can also test the complete application, even testing the updates in an integrated way, with the rest of the application.

6. Future work

In this section we propose some of the possible improvements that can be made to the idea and to the prototype of the environment we propose.

We can consider the actual limitations of the proposed idea and the developed prototype, and think about the possible horizon in the development of new programming environments on the web, completely integrated.

The possible improvements we list here, are not included in the work already done, but could be very useful to include them in the future, in order to provide more benefits and allow the team work in a development team, even if the members of that team are working from different physical places.

6.1 Multiuser features, for the multiuser team work

The environment allows the edition at the same time by different users, but in the developed prototype it is not indicated if another user is updating some file.

This can bring some difficulties. For example, a user can be working on a file version that is not the most recent update, and then, when he saves the file, he can be overwriting the file modified by another user.

The improvement we propose is similar to the feature included in Google docs, in which the environment indicates which user is working on a document.

Another possible improvement to favour the team work is the possibility of having users with different access rights in the project. For example, an administrator/creator user can edit any file, add new users to work on the project, and assign the rights for every user.

Suppose that we want to work with a graphical designer to do the graphical user interfaces, but we do not want this user to access Java code or access certain internal data, which may be sensible information in the business.

In this case, the administrator user can create a new user and give him rights to update only .jsp and .css files, avoiding giving him rights to the use of PhpMyAdmin in orden to protect data.

A chat feature could be included, to make possible the communication among different users in different locations. This can avoid the users to use another chat application such as MSN or GTalk to exchange messages releated to their work.

6.2 Awareness mechanism features

Awareness mechanisms could be included in the environment. This means, perception mechanisms that allow to have meta-information about the other users in a team. Those can be other forms of interaction among users.
The awareness mechanisms are very important to the collaborative work, especially when team members are in different locations.

It could be added: user interface design, interruption management (by other members of the team), work sincronization, definition of the information to be presented to the users such us the tasks other members are working on, etc.

6.3 Change management

With change Management features, the work done by programmers can be monitored with an existing change management system (tracking) such as Jira.

The feature could include the definition of issues related to the project and assign them to the different users. Each user could visualize and make comments on them.

This functionality could be very useful for and administrator or manager of the project, because it allows to have many information about the state of the project, the work done by any user and the pending tasks.

6.4 Additional reports

There are other reports apart from the provided in the actual prototype, that could be useful to have in the environment. For example, we could add reports about :

- quantity of code lines added by user
- quantity of hours each user was logged into the environment
- completed issues by user
- pending issues by user
- quantity of hours the project can be delayed considering pending issues
- etc

6.5 Software quality process features

It would be very useful to add some features for the software quality assurance in projects. For example, the measurement of metrics related to the tasks and processes, useful for developers, leaders and managers of the project.

6.6 Improvements in the file edition feature

It would be very useful to provide some facilities to the user in the edition of files. This could improve the work of a programmer by optimizing his resources.

Practicity and performance are very valuable features for programmers, and this is one of the main problems in the edition of texts on the web.

6.7 Possibility to upload .war files

The user can import a project built in another environment, outside JOnE environment. In this case, we could add a new feature that allows to upload a .war file allowing to build a project in JOnE with the files imported from the external application. This new feature requires that the .war file contains the source code (which is not mandatory in the environment).

7. Conclusions

Nowadays, there is a tendency to replace desktop applications with modern applications on the web. This is true for every software applications, from enterprise applications and information systems, to instant message and entertainment applications.

Nevertheless, up to now, this evolution has not achieved software development environments and developer tools. For example, database administrators do not allow to have web clients.

Web IDE environments, such as the one proposed in this work, are not developed yet, they are in the very first steps, or they are oriented to certain functionalities (such as code edition).

Besides, web interfaces of application servers do not provide all the functionalities provided by the desktop environments.

JOnE provides a new idea in the context of development environments. In the implemented version of JOnE, we could build a web environment that allows the programmer and other team members, to develop complete and integrated applications, completely on the web.
Users do not need to have installed applications or development environments in their computers, but they can access their projects from anywhere.

If we consider the functionalities proposed as future work, with JOnE we are providing the whole development team work a powerful tool. Examples of those functionalities include: version control, database administration, advanced code editor, collaborative work features, etc. All those functionalities could be integrated in JOnE, improving and extending even more the functionalities and benefits it provides.

Some of those functionalities already exist, but they could be integrated in a unique web environment and allow multidisciplinary team work (including programmers, graphics designers, database administrators, etc.). As a result we could have better workflow and improved software quality.

Through the use of plugins, we could offer support for many software quality model. This way, the users do not have to use certain applications or tools to apply quality policies

With the advance of web technology, for example in the area of JavaScript applications and the support provided by tools such as Ajax, JOnE provides a dynamic way of programming such as the obtained by desktop applications, but on the web.

Analyzing the tendencies in software development and the use of Internet for almost every activity, it is possible that in the near future we see programmers developing applications in web IDEs such as JOnE. In this sense, we can consider this work as a very important contribution for the information systems community.

References

(1) Bryan Basham, Kathy Sierra, Bert Bates. Head First Servlets and JSP. Second Edition. O'Reilly Media
(2) Source code and documentation about YUI: http://developer.yahoo.com/YUI/
(3) Information about JEE y Java Technologies : http://Java.sun.com/reference/index.jsp
(4) Documentation about Apache Tomcat: http://Tomcat.Apache.org/Tomcat-6.0-doc/index.html
(5)) Documentation about MySQL: http://dev.MySQL.com/doc/refman/5.1/en/
(6)) Documentation about Ubuntu: https://help.Ubuntu.com/9.04/index.html
(7)) Documentation and general information about de Apache Web Server: http://httpd.Apache.org/docs/2.2/
(8)) Documentation about PHP: http://www.PHP.net/docs.PHP