

JOnE: JEE Online Development Environment Programación en la Nube

Pablo Larrimbe , Pablo Salgado , Vanesa Maiorana

UADE - Universidad Argentina de la Empresa
Facultad de Ingeniería y Ciencias Exactas,
Ciudad Autónoma de Buenos Aires, Argentina
[pablolarrimbe,pdsalgado,vanesamaiorana]@gmail.com

Abstract. En este trabajo se presenta una propuesta novedosa para el desarrollo de aplicaciones, denominado JOnE (JEE Online Environment): un ide o entorno de desarrollo web para Java. Mediante el uso de la idea propuesta y la herramienta desarrollada, se podría programar una aplicación web completa sin tener una máquina con el ambiente de desarrollo instalado y configurado. Simplemente, con una conexión a Internet y un browser, el equipo de desarrollo podría construir el sistema y modificarlo en forma colaborativa desde cualquier lugar. El objetivo fundamental de este trabajo ha sido alcanzado: sentar un precedente a lo que se encuentra a la vanguardia en entornos de programación; no quizás por su gran complejidad o su avanzada funcionalidad, sino por el cambio de enfoque planteado. En el presente trabajo, se presenta la idea del nuevo enfoque de desarrollo, y también se demuestra su validez a través de la herramienta implementada, la descripción de su funcionamiento, sus ventajas y sus limitaciones.

1 Objetivos

El objetivo del trabajo es hacer que el proceso de programación esté al alcance de todos los usuarios, independientemente de las herramientas que éstos tengan instaladas. De ahí la razón para hacer un entorno web que integre todas las herramientas necesarias para cualquier desarrollo web. El objetivo principal del trabajo es que el usuario se desentienda de la instalación y configuración del software que sirve de soporte a la aplicación y de los problemas de infraestructura que puedan llegar a existir. El servidor web, la base de datos y el ide a utilizar se encuentran del lado del servidor. Lo único que necesita hacer el usuario es programar la aplicación. Luego, el entorno se encarga de compilar el proyecto, administrar el servidor web y la base de datos, haciendo todo esto de manera transparente y amigable para el usuario, teniendo en cuenta ciertas limitaciones relacionadas a la tecnología web hoy en día respecto de las aplicaciones de escritorio (desktop). Para complementar el objetivo descripto, vale aclarar en qué consiste el desarrollo mencionando los pasos que debería seguir un usuario que desea crear su aplicación web. Primero, deberá crear un usuario o ingresar al sistema. Luego podrá crear un nuevo proyecto o seleccionar un un proyecto ya

creado. Luego puede crear los archivos del proyecto, que pueden ser: Java Script, JSP, Css, Java y Servlet. El usuario también podrá organizar sus archivos a través de carpetas y así poder crear la estructura de la aplicación. Podrá por otro lado, compilar el código ingresado cuantas veces quiera y el sistema indicará cada vez los errores encontrados en la compilación del archivo. Una vez listo el proyecto, el usuario lo podrá desplegar ("hacer deployment"), descargar el archivo .war y finalmente probar la aplicación funcionando.

2 JOnE: Ambiente de programación en la nube

JOnE surgió del análisis de necesidades de un desarrollador para construir una página web en Java. Estas necesidades involucran, entre otras cosas, instalar un ide (actualmente los ides son herramientas de escritorio) como Eclipse, instalar una base de datos, instalar un servidor web como Tomcat, dedicar un equipo con cierta capacidad y condiciones acorde a lo requerido, etc. JOnE es una herramienta integrada de desarrollo (Integrated Development Environment) que funciona via web. El ide permite realizar las diversas tareas involucradas en un proyecto de desarrollo de software: la codificación propiamente dicha, el manejo de la persistencia (base de datos relacional), el testeo, y el deployment. El usuario puede crear, modificar y/o eliminar proyectos web Java. En un proyecto web Java se incluyen archivos Java Script, .css, Servlets, .jsp, .java, librerías (archivo .Jar) y un archivo descriptor de la aplicación web (web.xml). El usuario puede editar los archivos mencionados y realizar upload de las librerías que necesita utilizar.

Cloud computing (computación en la nube de internet) es un paradigma novedoso que permite almacenar la información en servidores en Internet sin necesidad de contar con las aplicaciones instaladas en las máquinas locales de los clientes. Este paradigma se basa en el "software como servicio". Como ejemplos de Computación en Nube destacan Amazon EC2, Google Apps, eyeOS y Microsoft Azure, que proveen aplicaciones comunes de negocios en línea accesibles desde un navegador web, mientras el software y los datos se almacenan en los servidores. Las aplicaciones en la nube son muy variadas, incluyen todo tipo de servicios a los clientes. Sin embargo, en el área de los entornos de desarrollo, aún hay mucho por hacer. Actualmente, no se han desarrollado cantidad de aplicaciones o herramientas parecidas al entorno propuesto en este trabajo. Algunas de ellas pueden ser categorizadas como editores de texto online. Estos son: Bepin, un proyecto que está en versión beta y es de Mozilla Labs; y Ecoder, un proyecto que también está en versión beta y es de Gmeditor. Ambos se pueden testear desde su sitio web de forma gratuita. El hecho de que ambos están en beta nos demuestra que aún es necesario un proceso de maduración de este tipo de aplicaciones para la industria del software. Existe otra aplicación, más parecida a un entorno de desarrollo, que también se encuentra en versión beta. La aplicación se llama WebDebStudio y fue realizada para codificar archivos y construir (hacer "building") de proyectos ANSI c y c++.

3 Estado del arte

Cloud computing (computación en la nube de internet) es un paradigma novedoso que permite almacenar la información en servidores en Internet sin necesidad de contar con las aplicaciones instaladas en las máquinas locales de los clientes. Este paradigma se basa en el software como servicio. Como ejemplos de Computación en Nube destacan Amazon EC2, Google Apps, eyeOS y Microsoft Azure, que proveen aplicaciones comunes de negocios en línea accesibles desde un navegador web, mientras el software y los datos se almacenan en los servidores.

Las aplicaciones en la nube son muy variadas, incluyen todo tipo de servicios a los clientes. Sin embargo, en el área de los entornos de desarrollo, aún hay mucho por hacer.

Actualmente, no se han desarrollado cantidad de aplicaciones o herramientas parecidas al entorno propuesto en este trabajo. Algunas de ellas pueden ser categorizadas como editores de texto online. Estos son: Bepin, un proyecto que está en versión beta y es de Mozilla Labs; y Ecoder, un proyecto que también está en versión beta y es de Gmeditor. Ambos se pueden testear desde su sitio web de forma gratuita. El hecho de que ambos están en beta nos demuestra que aún es necesario un proceso de maduración de este tipo de aplicaciones para la industria del software.

Existe otra aplicación, más parecida a un entorno de desarrollo, que también se encuentra en versión beta. La aplicación se llama WebDebStudio y fue realizada para codificar archivos y construir (hacer "building") de proyectos ANSI c y c++.

4 Funcionalidades de JOnE

En este apartado se describen las funcionalidades que están incluidas actualmente en JOnE:

4.1 Login

Con el acceso a través de su nombre de usuario y password, un usuario podrá acceder a los proyectos ya creados por él mismo y crear nuevos.

4.2 Administración de Proyectos

El usuario podrá agregar/editar/eliminar los archivos de su aplicación, dentro de proyectos. Además, podrá descargar archivos WAR, mediante el download del archivo estándar de una aplicación web JEE que podrá luego ser instalado en cualquier servidor web que siga el estándar de aplicaciones web JEE como Jetty o Tomcat. A través del administrador de proyectos, el usuario también podrá Administrar Tablas a través del editor de tablas de base de datos PhpMyAdmin, que se encuentra integrado a JEE Online Environment. Ver figura 1.

El usuario accederá directamente a su/s bases de datos con los permisos correspondientes. Hay una base de datos por cada aplicación, cuyo nombre es igual al de la aplicación.

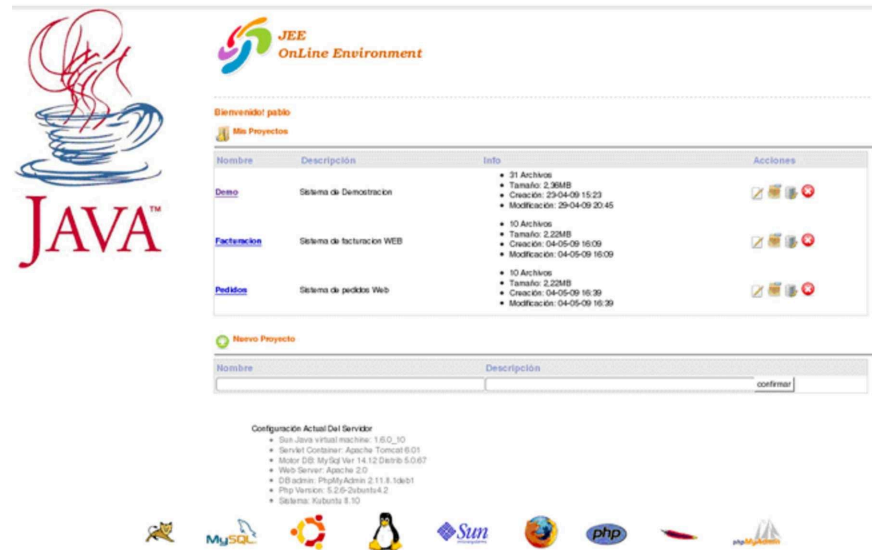


Fig. 1. JOnE. Pantalla de administración de proyectos

4.3 Administración de base de datos de la aplicación

JOnE está integrado a PhpMyAdmin (PMA) para la administración de bases de datos MySQL. Esta integración permite que sea transparente al usuario el cambio entre web Servers y además permite el login unificado de los usuarios. A través de esta funcionalidad, se pueden crear, editar y eliminar tablas, se pueden hacer consultas simples (tipo SELECT) o consultas más complejas (utilizando sentencias JOIN, UNION, etc). Ver figura 2.

4.4 Descargar war

Esta funcionalidad permite al usuario descargar el archivo .war para luego utilizarlo en su propio servidor, distribuirlo, modificarlo, etc. Ver figura 3.

4.5 Editor de código y compilador integrado

JOnE cuenta con un editor de textos para la edición de los archivos de una aplicación. El usuario podrá ver múltiples archivos a través de los tabs. En el

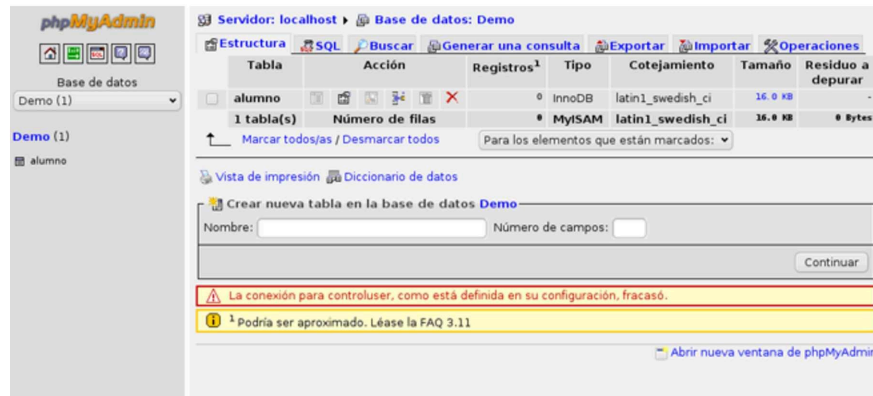


Fig. 2. JOnE: Pantalla de administración de bases de datos



Fig. 3. JOnE: Pantalla de descarga de archivos war

caso de archivos .java y .jsp, además de guardar el código fuente, JOnE realizará una compilación de los mismos. En el log se registrará la compilación satisfactoria o los errores de compilación obtenidos. La compilación es muy útil ya que no se debe esperar a hacer deployment de la aplicación en Tomcat y luego intentar ejecutarla para obtener los errores, sino que en tiempo de programación ya es posible hacer debug de la aplicación. El editor de texto indica qué archivo está siendo editado en cada momento, y posee como una ayuda el número de línea para revisar los errores de compilación (Ver figura 4).

4.6 Despliegue de la aplicación (deployment)

Esta funcionalidad incluye el despliegue de la aplicación construida. JOnE realiza el deployment en un servidor Tomcat, registrando cualquier inconveniente en el archivo log (Ver figura 5).

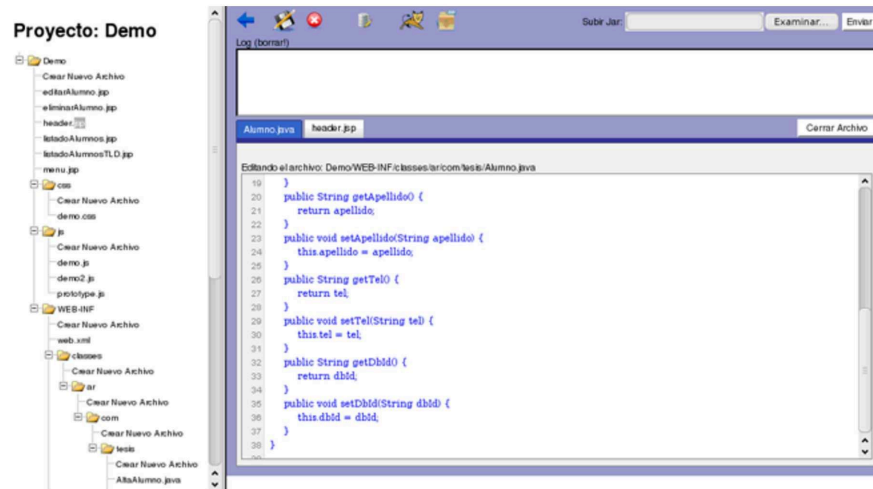


Fig. 4. JOnE: Editor de código

Para archivos .java el usuario podrá elegir si desea que se guarden como Servlets. Al elegir esta opción el sistema actualizará el descriptor web.xml de forma automática agregando las directivas estándar. El archivo web.xml es editable por parte del usuario (como un archivo más de la aplicación) para poder personalizar las directivas.

El archivo log indica si los archivos pudieron ser guardados y (según corresponda) compilados correctamente o los errores de compilación obtenidos. Para Servlets, contiene la actualización del descriptor web.xml, si el archivo fue eliminado correctamente y si la aplicación pudo ser desplegada correctamente junto con el link para que pueda ser testeada. El log posee dos estilos diferentes. En color verde aparecerán las acciones que fueron cumplidas de forma exitosa y en rojo las acciones que fallaron.

4.7 Upload de archivo jar

Esta funcionalidad le permite al usuario incorporar librerías externas al proyecto web. En este caso la librería se colocará en la carpeta WEB-INF/lib del proyecto, siguiendo el estándar de una aplicación JEE web. Algunas librerías base son brindadas por la aplicación como lo son el soporte para Jslt (JSP Standard Tag Library) o el conector Jdbc de MySQL, que ya es compatible con la versión que se encuentra en el servidor.

4.8 Eliminar aplicación

Se pueden eliminar aplicaciones completamente, incluyendo la base de datos asociada a esa aplicación, los archivos físicos (desde el filesystem).

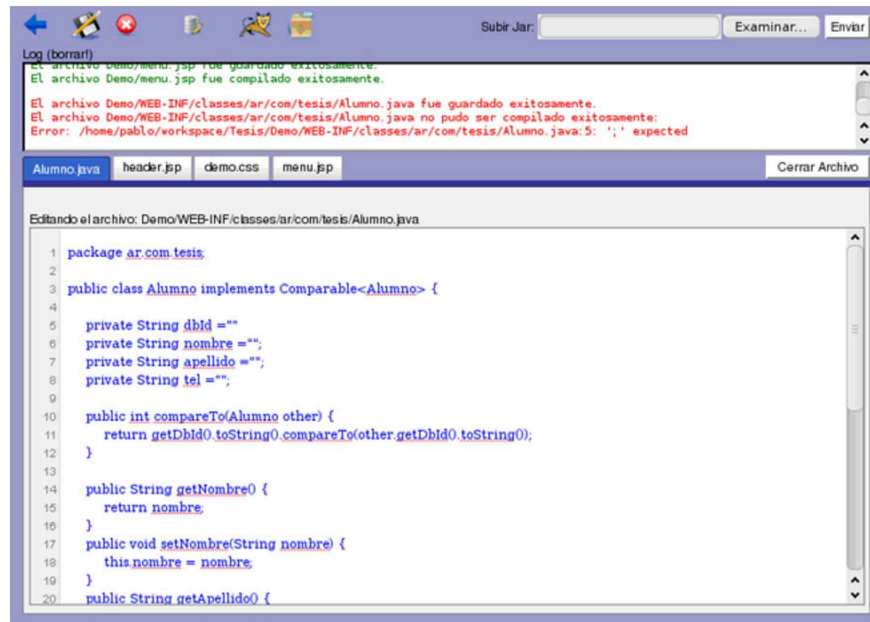


Fig. 5. JOnE: Visualización del archivo log

5 Ejemplo de una aplicación creada con JOnE

En la figura 6 se muestra una aplicación ejemplo, creada íntegramente usando JOnE como entorno de desarrollo vía web. La aplicación creada es una aplicación simple para administrar alumnos. En esta figura (Figura 6) se puede ver el listado de alumnos. Las opciones son: editar, eliminar, agregar un nuevo alumno, y listar alumnos. La opción del listado de alumnos fue desarrollada usando Taglibs Tld. Los Taglibs son una API para JSP en donde se colocan variables especiales en el archivo que luego serán reemplazadas dinámicamente por datos reales, en este caso, los alumnos.

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
	Rodigo	Lopez	Editar	Eliminar

Fig. 6. Aplicación construida con JOnE en ejecución

Veamos ahora un escenario real donde se ve el beneficio del uso de JOnE: supongamos que queremos cambiar la interfaz de usuario (look and feel) del

administrador (ABM) de alumnos. Si el usuario deseara cambiar la interfaz, y por ejemplo, la misma fue construida con una plantilla por un diseñador gráfico, éste tendría que instalar la aplicación que le permita modificar esta plantilla. Esto implica que, sin la utilización de JOnE, debería obtener las librerías a ser utilizadas, instalar la base de datos de la aplicación junto con el servidor MySQL, instalar Java, instalar un servidor Tomcat para poder realizar las modificaciones pertinentes y para poder probar la aplicación funcionando antes de entregársela al cliente. En cambio, mediante el uso de JOnE, el diseñador podría acceder directamente solo utilizando un browser y modificando el archivo .css, .js, .html o .jsp correspondiente, para luego poder probar fácilmente la aplicación. Veamos en detalle lo que debería hacer el diseñador si tuviera que cambiar, por ejemplo, el color del menú:

- 1) Ingresar a la aplicación, con su nombre de usuario y clave.
- 2) Abrir los archivos correspondientes: en este caso es el archivo demo.css y modificar el estilo que corresponda.
- 3) Para asegurarse que los estilos han quedado bien, debe hacer deployment de la aplicación y entrar a la url que el archivo log generado por JOnE le indica.
- 4) Comprobar que los estilos están bien y nos dice que el menú ya está con el estilo correspondiente.

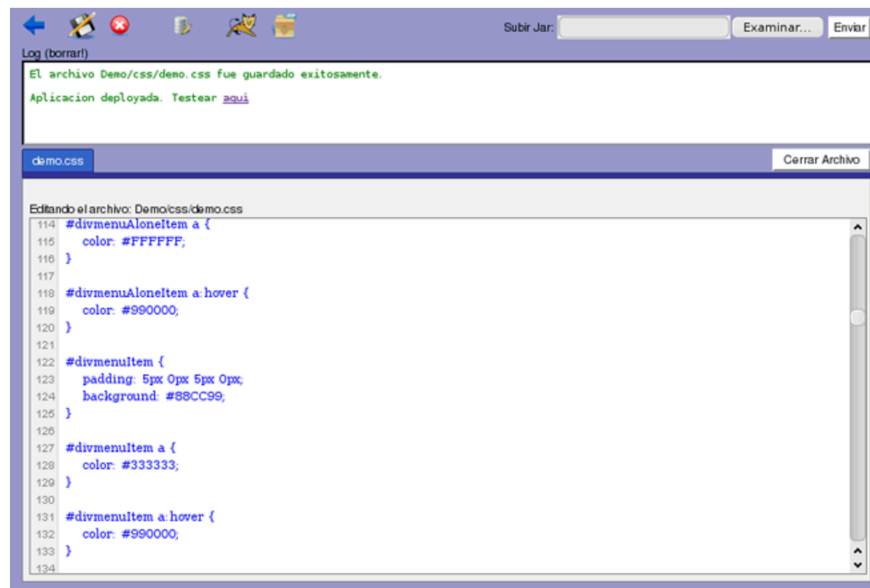


Fig. 7. Edición de plantilla de diseño en la aplicación ejemplo

La figura 7 representa los pasos 2 y 3. El diseñador edita los estilos (demo.ccs) de la aplicación, en este caso, edita el color de fondo de los ítems del menú. Luego

hace deployment de la aplicación. Aquí es posible observar además otra particularidad del archivo log. En el caso de que la acción realizada sea el deployment, se puede hacer clic dentro del log para que se abra una nueva pestaña en el browser o una nueva ventana (esto depende del navegador que se está utilizando) con la aplicación que se desea testear. Esto mejora la navegabilidad, ya que el usuario no debe abandonar JOnE para testear la aplicación sobre la cual está trabajando.

Menu		Listado de alumnos con jp 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. 8. Prueba de la aplicación ejemplo, luego de los cambios

La figura 8 representa al paso 4 donde el diseñador comprueba que efectivamente se han cambiado los estilos y que el menú cambió de color.

En este capítulo se ha presentado solo un ejemplo de uso de JOnE, donde se pueden aprovechar los beneficios de no tener que instalar todo el ambiente de programación. Los beneficios son numerosos, teniendo en cuenta que son muchos los escenarios posibles de modificación y actualización de una aplicación. Si se trata de proyectos de gran envergadura, el equipo de desarrollo estará conformado por varias personas con diferentes perfiles. Usando JOnE, cada persona accederá y modificará lo que necesita, pudiendo probar la aplicación completa, incluso testeando la integración de sus modificaciones con el resto de la aplicación.

6 Trabajos futuros

En esta sección se presentan algunas posibles mejoras a la idea y entorno propuestos, tratando de comprender cuales son los límites actuales y hasta donde se puede llegar con la construcción de entornos de programación web completamente integrados. Estas posibles mejoras no están incluidas actualmente en el desarrollo realizado, pero serían de gran utilidad para ser agregados en el futuro y así proveer aún más beneficios al trabajo de desarrollo de software en equipo y a distancia.

6.1 Facilidades para el trabajo multiusuario

El entorno permite la modificación en simultáneo por parte de diferentes usuarios, pero actualmente no se indica si otro usuario está editando algún archivo, ni quién es este usuario. Esto puede traer algunas complicaciones, como por ejemplo, que se está trabajando sobre una versión que no es la última y luego al grabar se sobre-escriba el trabajo de otro. Esta funcionalidad sería similar

a la que provee Google docs, en el cual se le indica a los usuarios que tienen visibilidad sobre el documento, que se están editando los archivos.

Otra posible mejora para favorecer al trabajo en equipo sería la posibilidad de tener usuarios con diferentes permisos sobre el proyecto. Por ejemplo, que exista un usuario administrador/creador del proyecto que pueda, además de editar cualquier archivo, asignar nuevos usuarios al proyecto y los permisos que posee cada uno de estos usuarios. Supongamos que queremos trabajar con un diseñador gráfico para el diseño de las interfaces de usuarios, pero no queremos que éste pueda tener acceso al código Java o a ciertos datos internos, ya que es información sensible para el negocio. En este caso, el administrador de la aplicación puede crear un usuario y otorgarle permisos solo para los archivos .jsp y .css y quitarle permisos para el uso de PhpMyAdmin, para proteger los datos.

Se podría también incluir un chat en el cual participen los usuarios de los proyectos con la finalidad de que los usuarios no tengan que utilizar otra aplicación como MSN o Gtalk para intercambiar mensajes relacionados al trabajo.

6.2 Mecanismos de awareness

Se podrían incluir mecanismos de awareness, es decir, mecanismos de percepción que permitan tener meta-información sobre los demás usuarios, formas en las que llevará a cabo la interacción entre ellos, etc. Estos mecanismos de awareness son muy importantes para el trabajo colaborativo entre personas que están distantes físicamente. Podrían incluir diseño de interfaces de usuario, manejo de las interrupciones por parte de otros miembros del equipo, sincronización del trabajo, definición de la información a ser presentada a los usuarios sobre lo que están haciendo los demás miembros del equipo, etc.

6.3 Seguimiento de cambios

Lo que se quiere lograr con esto es que el trabajo de los programadores pueda ser monitoreado con algún sistema de seguimiento de cambios (tracking) como por ejemplo, Jira. Se podrían cargar issues relacionados al proyecto, asignárselos a los usuarios y que cada uno de ellos pueda visualizarlos y realizar comentarios. Esta sería una funcionalidad clave para un administrador del proyecto, ya que podría obtener mucha información sobre el avance del mismo y las cuestiones pendientes.

6.4 Reportes adicionales

Se podrían agregar reportes sobre cantidad de líneas de código agregadas por usuario, por ejemplo, sobre las horas en las cuales cada usuario estuvo conectado, issues completados o pendientes de cada usuario, horas que demorará el proyecto en base a los issues cargados, etc.

6.5 Procesos de calidad de software

Procesos de calidad de software: sería de gran ayuda agregarle a JOnE procesos de calidad de software, posibilidad de medición de métricas, teniendo en el mismo entorno, todas las tareas o procesos que engloba el desarrollo, para desarrolladores y líderes de proyecto.

6.6 Mejoras en la edición de archivos

Se podrían brindar comodidades al usuario para la realización de la edición de archivos, intentando mejorar la dinámica que requiere un programador con la finalidad de optimizar sus recursos. Se deberá atender a la practicidad y a la performance (gran problema de la edición de textos en web).

6.7 Posibilidad de upload de archivos war

La idea es que el usuario pueda importar un proyecto que tiene en otro entorno, a JOnE. Se implementaría la funcionalidad de upload war en donde el usuario podría subir un archivo del tipo war y JOnE armaría un proyecto con los archivos de esa aplicación. Para la realización de este ítem se requeriría de que el war contenga el código fuente (no obligatorio por defecto).

7 Conclusiones

Actualmente, se tiende a reemplazar las aplicaciones de escritorio por más modernas aplicaciones web. Esto sucede en todos los niveles de software, desde aplicaciones empresariales y sistemas de información, hasta sistemas de mensajería instantánea y de entretenimiento. Sin embargo, hasta el momento no se ha dado esta evolución en el área de los entornos de desarrollo de software y las herramientas para desarrolladores. Por ejemplo, las aplicaciones como administradores de bases de datos no permiten tener clientes web.

Los entornos ide web, como el propuesto en este trabajo, aún no se han desarrollado, están en sus etapas iniciales o están orientados a funcionalidades parciales (como la edición de código). Asimismo, las interfaces web de los servidores de aplicaciones, no proveen todas las funcionalidades que sí proveen sus equivalentes de escritorio. En ese sentido JOnE da un paso hacia adelante en el contexto del desarrollo de software.

En la versión implementada, se ha logrado un entorno de programación web que le permite al programador, desarrollar aplicaciones en forma completa e integrada, desde la web. No necesita tener aplicaciones ni entornos instalados en su máquina de escritorio, y puede acceder a sus proyectos desde el lugar donde se encuentre.

Si tenemos en cuenta las futuras mejoras presentadas en este trabajo, se estará brindando al equipo entero de desarrollo una herramienta completa e integrada. Ejemplos mencionados incluyen facilidades de seguimiento de cambios,

controlador de versiones, administrador de bases de datos, editor avanzado de código, etc. Todos ellos se podrían integrar a JOnE ampliando aún más las capacidades de las que ya posee. Estas herramientas ya existen actualmente en el mercado, pero JOnE integraría todas ellas en un entorno web y permitiría el desarrollo en equipos de trabajo multidisciplinarios (programador, diseñador gráfico, administrador de bases de datos, etc.) agilizando el flujo del trabajo y mejorando la calidad de las aplicaciones.

Mediante el uso de plugins se podría dejar la puerta abierta a brindar soporte a cualquier modelo de calidad que se desee implementar, con la posibilidad de cumplimentar algunos flujos de información propios o algún tipo de documentación específica de cada modelo de calidad en particular. De este modo el usuario no está obligado a utilizar una herramienta adicional para respetar las normas de calidad, sino que el modelo se integraría con JOnE. Con el gran avance que ha tenido la tecnología web, como por ejemplo, en materia de aplicaciones hechas en Java Script y con el soporte brindado a herramientas como AJAX, JOnE logra un dinamismo similar al de una aplicación desktop, pero en la web.

Analizando las tendencias en el desarrollo y en el uso de Internet para todas las actividades, es posible que en algunos años los programadores estén desarrollando aplicaciones utilizando ides web como JOnE. En este sentido podría considerarse este trabajo como un aporte muy importante a la comunidad informática.

8 Bibliografía

- (1) Bryan Basham, Kathy Sierra, Bert Bates. Head First Servlets and JSP. Second Edition. O'Reilly Media
- (2) Código fuente y documentación sobre YUI: <http://developer.yahoo.com/YUI/>
- (3) Información sobre JEE y tecnología Java: <http://Java.sun.com/reference/index.jsp>
- (4) Documentación sobre Apache Tomcat: <http://Tomcat.Apache.org/Tomcat-6.0-doc/index.html>
- (5) Documentación acerca MySQL: <http://dev.MySQL.com/doc/refman/5.1/en/>
- (6) Documentación sobre Ubuntu: <https://help.Ubuntu.com/9.04/index.html>
- (7) Documentación e información general sobre de Apache web Server: <http://httpd.Apache.org/docs/2.2/>
- (8) Documentación acerca de PHP: <http://www.PHP.net/docs.PHP>