

Hacia los IDE de siguiente generación: Revisión de herramientas de desarrollo.

Annie Patterson¹, Mirna Víquez¹, and Externo a la ULACIT²

Escuela de Ingeniería,
Universidad Latinoamericana de Ciencia y Tecnología,
ULACIT, Urbanización Tournón, 10235-1000
San José, Costa Rica
apattersonr362@ulacit.ac.cr, mviquez948@ulacit.ac.cr
<http://www.ulacit.ac.cr>

Abstract. La utilización de un entorno de desarrollo integrado (IDE) requiere la instalación de un conjunto de aplicaciones y constituye una de las herramientas más importantes para los programadores, debido a que les proporciona los elementos necesarios para el desarrollo de sistemas de software. Sin embargo, las necesidades de apoyar a los programadores con ambientes de trabajo más flexible apuntan a la adopción de los servicios y características que ofrece la “Computación en la Nube”. Esta tendencia se está viendo materializada en el uso de “IDEs online”, para los cuales no es requerido realizar instalaciones y configuraciones en equipos o dispositivos, y solo requiere contar con conexión a Internet y un navegador web. De acuerdo con lo anterior, este trabajo lleva a cabo un análisis de siete IDEs en la nube que se encuentran disponibles de forma comercial y gratuita. Dicho análisis tiene como fin la identificación de las principales características de estas herramientas de acuerdo con los lenguajes de programación que soportan, tipos de licenciamiento que ofrecen y las herramientas SCM y de Bug Tracking Tools con las cuales se pueden integrar.

Keywords: Computación en la Nube, IDEs, IDEs Online, SCM.

1 Introducción

Las herramientas que existen en la actualidad para desarrollar software son accesibles y fáciles de utilizar, pero en muchos casos dependen de un determinado sistema operativo, y en el caso de hardware, por sus características físicas (e.g. peso y dimensiones) la movilidad de los usuarios sufre limitaciones. Por lo que la tendencia es mover las herramientas a ambientes de alta disponibilidad en la *nube*, para permitir a los desarrolladores efectuar tareas de programación desde cualquier equipo y localidad geográfica, mientras hace uso de su entorno personal de trabajo.

Entre las ventajas que ofrece el modelo de programación en la nube se encuentran las siguientes (Murazzo1, n.d.-a):

1. No es necesario instalar herramientas para cada usuario
2. Las actualizaciones y mantenimiento de los sistemas se realizan al entorno de trabajo que utiliza un gran número de usuarios
3. Los programadores cuentan con entorno de ejecución implícito [5].

En este contexto se debe tener en cuenta que las empresas cuentan con empleados distribuidos en diferentes regiones y países, y que el uso de las tecnologías de información es indispensable para la comunicación entre los miembros de los equipos de trabajo, así como para el buen funcionamiento de las organizaciones. Por lo que es necesario implementar plataformas tecnológicas para apoyar el trabajo colaborativo de los usuarios, sin importar su ubicación geográfica. Por lo que los servicios que ofrece la computación en la nube permite contar con un entorno que facilita la colaboración a la vez que se adapta a las demandas no previsible o de picos de trabajo, y las organizaciones solo deben pagar por el consumo efectuado [1].

De acuerdo con lo anterior, los objetivos de esta investigación son identificar, analizar, valorar y comparar los principales entornos de programación (IDE), así como herramientas para la administración de versiones de código (SCM) y gestión de fallas (Bug Tracking), basadas en la nube, tanto comerciales como *open source*. Esto con el fin de conocer sus características, especificaciones, ventajas y desventajas.

2 Antecedentes

La programación en la nube ofrece flexibilidad por medio de plataformas escalables para apoyar la alta demanda de recursos para un gran número de desarrolladores (Michael Armbrust & Matei Zaharia. María Murazzo1, 2010).

Entre las principales ventajas que ofrece el modelo de programación basado en la nube se encuentra el fomento de la colaboración en equipo (Cloud-Based Software Crowdsourcing) , mayor capacidad de procesamiento, el ahorro de espacio de almacenamiento, y la reducción de costos (DESARROLLO DE APLICACIONES PARA CLOUD COMPUTING). Sin embargo, conviene considerar que la nube no diferencia entre información común e información sensible, existe un riesgo latente de robo de información porque que muchos proveedores alquilan servidores de terceros para reducir costos, y tanto programadores como usuarios desconocen esta situación. Por lo que se pueden presentar problemas de privacidad de la información. De forma adicional conviene considerar que si el proveedor de los servicios de la nube decide cerrar los servicios, puede presentarse la pérdida de los datos sin posibilidad de recuperación.

Es importante que los desarrolladores de software comiencen a incursionar en el mundo de la programación en la nube. Esto le dará más valor en el mercado laboral, además de la importancia de mantenerse actualizado en las nuevas tecnologías. Tanto el programador como el proveedor, deben ser responsables de la forma en que manejan los datos dentro de la nube; para evitar inconvenientes es recomendable que el programador no se apegue completamente a la nube y

realice sus respaldos del trabajo realizado y busque seguridad adicional para mantener la integridad de sus datos.

2.1 Personajes que promovieron el concepto de la computación en la Nube

Lo que hoy conocemos como "Computación en la Nube", y las ventajas que nos proporciona, posee un gran legado de aportes, inspirados por visionarios que tuvieron una perspectiva de futuro con respecto a la utilidad de la computadora. Entre los personajes que formaron parte de este gran legado, se destacan los siguientes:

- En 1950 Herb Grosch, definió la Ley de Grosch: "La potencia de la computadora es proporcional al cuadrado de su precio", en lo que algunos investigadores la relacionan al concepto de la computación en la nube, ya que argumentaba que las economías eficientes y adaptables podría alcanzar su objetivo si confían en centros de datos centralizados en lugar de confiar en el almacenamiento de unidades.
- Alrededor del año 1961, John Patrick McCarthy se le atribuye la definición del concepto de la nube al matemático, informático y profesor ya difunto, en donde declaró públicamente en un discurso el concepto de tecnología de tiempo compartido (Time Sharing) de las computadoras y sus servicios, que serán obtenidas y pagadas por consumo, y pone de ejemplo el consumo del agua y la luz.
- En el año de 1996, Douglas Parkhill mediante la publicación de su libro llamado "El desafío de la utilidad de la computadora", pone en manifiesto las principales características que actualmente conocemos de la computación en la nube, uno de los principales "pago por consumo", para ello, en su libro hace la comparación de las características de la computación en la nube con la industria eléctrica.

2.2 Empresas pioneras en la computación en la Nube.

De igual forma, en el transcurso de los años varias empresas apostaron sobre los beneficios de implementar este tipo de tecnologías, que progresivamente fueron reconociendo las infinitas ventajas y cambiando radicalmente la forma de adquirir los servicios, así como la forma de trabajo de sus colaboradores, a continuación se mencionaran algunas de las empresas que fueron pioneras en brindar servicios basados en la nube, esto según lo analizado por Cruz Valencia, Katherin en Historia Del Cloud Computing. RITS [online]. 2012, n.7 :

- Salesforce.com, en 1999, introdujo el concepto de entrega de aplicaciones empresariales a través de una sencilla página web.
- Amazon al publicar Amazon Web Service en el 2002.

- Google Docs en el 2006.
- Elastic Compute Cloud de Amazon (EC2) en el 2006, como un servicio web comercial que permitió a las empresas pequeñas y particulares alquilar equipos en los que pudieran ejecutar sus propias aplicaciones informáticas.
- Eucalyptus en el 2008, como la primera plataforma de código abierto compatible con el API-AWS para el despliegue de clouds privados.
- OpenNebula, el primer software de código abierto para la implementación de nubes privadas e híbridas.
- Microsoft en el 2009 con el lanzamiento de Windows Azu
- Apple en el 2011, Apple lanzó su servicio iCloud, un sistema de almacenamiento en la nube - para documentos, música, videos, fotografías, aplicaciones y calendarios que prometía cambiar la forma en que usamos la computadora.

2.3 IBM innova con Bluemix.

Lanzado en su versión beta en febrero del 2014, es reconocida como unas de más destacadas aplicaciones basadas en Cloud Foundry, opera bajo la infraestructura de SoftLayer y bajo la ideología de producir productos y servicios de software fácil y rápidamente. Es preciso aclarar que Cloud Foundry es una PAAS open Source que brinda VMWARE, caracterizada por ser una plataforma opensource multi-lenguaje, multi-framework, multi-aplicación y multi-cloud.

Dicha plataforma permite la utilización de varios frameworks y servicios, que le facilita el trabajo a los desarrolladores, por lo que solo es requerido especificar con cuáles de ellos es necesario utilizar. En la tabla 1, se especifica algunos de ellos.

Frameworks	Servicios
Spring	MySQL: Base de Datos Relacional.
Ruby on Rails	Vfabric Postgres: Base de Datos Relacional.
Ruby and Sinatra	MongoBD: Base de Datos No Relacional.
Node.js	Redis:Base de Datos Key-Value.
–	RabbitMQ:Gestor de Colas de Mensaje

Table 1. Frameworks que soporta Bluemix.

Bluemix da toda la plataforma para el desarrollo de aplicaciones Web en Ruby, PHP y JAVA, en el caso de aplicaciones para móviles maneja iOS, Android y HTML con JavaScript sin que los desarrolladores se preocupen del hosting ni la infraestructura que es requerida para gestionar y alojar el sitio.

3 Desarrollo

Un IDE, o entorno de programación, se define como la aplicación que se utiliza para editar, compilar, depurar y construir la interfaz gráfica de nuestro código de una forma amigable.

En un inicio estas aplicaciones se podían utilizar solamente a nivel de escritorio, pero conforme ha evolucionado la forma de almacenar la información se ha comenzado a migrar dichos entornos a sus propias versiones basadas en navegadores web con las mismas características que los basados en escritorio.

A continuación mencionaremos los principales entornos utilizados y su compatibilidad con los lenguajes de programación más comunes en la actualidad (Higuera & Ricaurte., 2011).

IDE	HTML/ HTML5	CSS	JS	Ruby	PHP	Phyton	Java	C/ C++	C#	Android	Node.js
Codio	X	X	X	X		X	X	X			
Codenvy			X	X	X	X				X	X
Cloud9			X	X	X	X		X			X
Codebox	X			X	X		X	X			X
CodeAnyware	X	X	X		X						
Compilr	X			X	X	X		X	X		
Orion	X		X								

Table 2. IDEs y su compatibilidad con los lenguajes de programación más comunes.

(Mwootendev, 2011)

En la Tabla 2 Se muestra el cuadro comparativo de IDEs con compatibilidad a lenguajes de programación (Kaplan, 2014). Podemos notar como en su mayoría se enfocan en lenguajes de programación web (HTML, PHP, Ruby y JavaScript), pero también buscan su adaptación con lenguajes de escritorio y móviles comunes como C#, C/C++, Java, Android. Por lo que podemos determinar que la programación en la nube no está limitada a una sola plataforma.

Cabe indicar, que estos entornos se adecuan a las posibilidades económicas de los desarrolladores, todos tienen su versión gratuita y tienen diferentes formas de licenciamiento, en su mayoría poseen opciones gratuitas, en los que se destacan Codio, Codenvy (Codenvy, 2014), Cloud9 (Cloud9, 2014), Complir (Compilr, 2014), CodeAnywhere y Orion (Orion, 2014), pero también proporcionan planes de pago en lo que se requieren trabajar en proyectos de códigos privados, como se puede apreciar en la tabla 3:

Como observamos en el cuadro anterior, lo único que restringe los planes de paga es la capacidad de almacenamiento, pero en su mayoría las características principales se pueden utilizar desde la versión gratuita. Dentro de todo programa de software existen pulgas o “bugs” que se refiere a los problemas que se puedan encontrar en el mismo durante el tiempo de ejecución del mismo. Por esta causa,

IDE	Planes y sus costos
Codio	Para privados \$8 al mes.
Cloud9	Con un plan de \$19 se pueden obtener más opciones que la versión gratuita.
Codebox	La versión Premium permite 10 proyectos a \$9 por mes.
Codeanywhere	Tiene los tipo Starter y Frelancer:El plan mensual tiene un costo de \$3 hasta \$25 mensuales

Table 3. Los Planes que ofrecen IDEs.

debemos contar con un sistema que nos ayude a rastrear y corregir dichos problemas. Estos sistemas se conocen como sistemas de BugTrack, los cuales permiten notificar dichos problemas de forma centralizada, estableciendo un flujo de trabajo basado en un estado a través de los “bugs”, creando así un historial de bugs (de Sevilla., 2013).

En la tabla 4 presentada a continuación, mostraremos los principales sistemas de BugTrack y sus características:

En la tabla siguiente, se muestra las cuatro bases de datos más utilizadas por los entornos de desarrollo de estudio, siendo “MySQL”, la más soportada para todos los casos, exceptuando las herramientas Compilr e Orion, para las que no se logró determinar cuáles bases de datos soporta o qué tecnologías incorpora para estos fines.

Nombre	Características
Bugzilla	Busca "bugs" y cambios en los códigos. Habilita una comunicación rápida con otros compañeros de trabajo. Establece y revisa "parches". Basado en la Web. Posee una sencilla interfaz. De código abierto y de uso gratuito.
Jira	- Guarda y presenta informes de flujo de trabajo. - Se integra con otros entornos de desarrollo de código. - No solo se centra en la programación, también funciona para "help desk", renderizado entre otros. - Es de licencia comercial.
Mantis	- Basado en la Web. - Interfaz sencilla. - Posee su versión móvil. - Licencia gratuita. - Implementada en PHP.
Trac	- Desarrollado en Python. - Basado en la Web. - Utilizado con una herramienta SCM, se puede navegar entre el código, ver cambios, entre otros. - Los problemas encontrados se almacenan como tickets.
Redmine	- De código abierto. - Se integra con herramientas SCM. - Es una aplicación Web.

Table 4. Principales sistemas de BugTrack y sus características

En todo proyecto de programación, en el que se involucre más de una persona, es requerido la utilización de SCM, "Source Code Management". Esto, para propiciar un ambiente de control, y haya un seguimiento adecuado de las versiones del código y de todos los archivos que se requieran compartir en el proyecto. Entre las herramientas basadas bajo el concepto SCM, desatacaron como las más utilizadas Git y su versión en línea Github, como pueden visualizar en la tabla 6.

4 Conclusiones

A pesar de la tendencia de las aplicaciones en la nube, la programación en este ambiente está apenas comenzando su desarrollo, por lo cual aún no es muy común que se utilice a nivel empresarial, aun cuando su costo comparado con la programación de escritorio es mucho más bajo (Murazzo1, n.d.-b).

Entre los aspectos más relevantes se rescatan los siguientes:

- Su compatibilidad con los principales lenguajes de programación modernos y bases de datos más comunes les da la potencialidad de llegar a ser la plataforma de programación en definitiva en los próximos años.

IDEs \ Bases de Datos	MongoDB	MySQL	PostgreSQL	SQLite
Codio	X	X	X	X
Codenvy	X	X		
Cloud9	X	X	X	X
Codebox		X		
CodeAnywhere		X		

Table 5. IDEs y Bases de Datos

IDEs \ SCM	Git	GitHub	Mercurial	Bitbucket	Subversion
Codio	X	X	X		
Codenvy	X	X			
Cloud9	X	X	X		X
Codebox	X	X		X	
CodeAnywhere		X		X	
Orion	X	X			

Table 6. IDEs y Bases de Datos

- La tendencia de los IDE's más comunes de la nube es la programación web ya que lo que se busca en la actualidad es evitar la limitación del hardware para poder acceder a nuestras herramientas tecnológicas.
- Para todo tipo de desarrollo de software es importante contar con un sistema de rastreo de problemas que nos dé el detalle y hasta la posible solución de los problemas, dichas aplicaciones le ahorran al desarrollador horas de búsqueda dentro del código programado.
- Los IDE's basados en la nube más completos y utilizados para el desarrollo son Codio, CodeEnvy y Cloud9 ya que son los que más lenguajes soportan, pero no quiere decir que el resto no tenga buenas posibilidades ya que depende del tipo de software que se quiera desarrollar.
- La herramienta de BugTrack utilizada por excelencia es Bugzilla, pero al igual que los IDE's la escogencia del mismo depende de las necesidades del software.
- Codebox (FriendCode, 2014), Compilr y Codio, poseen la característica en común que son IDE's que poseen un enfoque educativo, por lo que fomentan el aprendizaje en línea a personas interesadas en incursionarse en el mundo de la programación. Ambas herramientas incluyen tutoriales y guías de estudio, lo cual facilita exponencialmente el aprendizaje por medio de cursos interactivos.
- todas las IDE de estudio solicitan la suscripción en su página oficial para poderla utilizar, independientemente del tipo de licenciamiento (comercial o

gratuito). De igual forma si no es deseado registrarse en directamente, se pueden utilizar las cuentas ya existentes de Facebook, Google Plus, Twitter, GitHub y LinkedIn, con el inconveniente que se estarían otorgando el acceso a toda la información personal contenida en dichas aplicaciones, situación que no ha sido de agrado por los usuarios.

- CodeBox y Codenvy poseen la particularidad que permite la opción de trabajar “offline”, para que en los casos que a los desarrolladores requieran acceder a sus proyectos y no posean en ese momento conexión a internet pueden hacerlo, y cuando ya poseen acceso a internet nuevamente, solo se debe iniciar la sincronización de los datos. A diferencia de CodeBox que utiliza la misma IDE para editar código tanto en el “escritorio” como en la “nube”, y Codenvy facilita esta opción “offline”, siempre y cuando se tenga instalado en el equipo “Eclipse Che”.

References

- Cloud9. (2014). *Cloud9 Inc.* Página web corporativa. pages 5
- Codenvy. (2014). *Codenvy Inc.* Página web corporativa. pages 5
- Compilr. (2014). *lynda.com Inc.* Página web corporativa. pages 5
- de Sevilla., U. (2013). Introducción a los sistemas de bugtrack. *Universidad de Sevilla*, 1, 1-21. pages 6
- FriendCode. (2014). *Codebox.* Página web corporativa. pages 8
- Higuera, F. H. L., & Ricaurte., J. A. B. (2011). Comparación de herramientas para el desarrollo de librerías enfocadas a aplicaciones web. *Universidad Católica del Norte, Revista Virtual*, 34, 342-359. pages 5
- Kaplan, G. (2014, jul). *Los 5 mejores ides para desarrollar en la nube.* Blog. pages 5
- Michael Armbrust, R. G. A. D. J. R. K. A. K. G. L. D. P. A. R. I. S., Armando Fox, & Matei Zaharia. María Murazzo1, N. R. D. S. D. V., Flavia Millán. (2010). A view of cloud computing. communications of the ac m. *Universidad Nacional de San Juan, De Google Scholar Base de datos*, 53, 50-58. pages 2
- Murazzo1, M. (n.d.-a).
pages 1
- Murazzo1, M. (n.d.-b).
pages 7
- Mwootendev. (2011, nov). *Coding in the cloud: Diving in with online ides.* Página Web. pages 5
- Orion. (2014). *The eclipse foundation.* Página web corporativa. pages 5