

Portafolios Docentes de Programación en la Nube para la Evaluación de Competencias

Programming Portfolios in the Cloud for Skills Assessment

Damià Segrelles, Germán Moltó, Franz Miranda
dqulils@dsic.upv.es, gmolto@dsic.upv.es, franz_miranda@outlook.com

Instituto de Instrumentación para Imagen Molecular (I3M)
Centro mixto CSIC - Universitat Politècnica de València
Valencia, España

Resumen- Este trabajo presenta una plataforma web para la generación de Portafolios Docentes de Programación Software (PDPs) en la nube y la extracción automática de métricas de los PDPs generados para evaluar competencias, tanto las relacionadas específicamente con la programación de software (p. ej. Desarrollar software de calidad, Dominar el paradigma orientado a objetos), como las competencias genéricas (p. ej. Capacidad de análisis y resolución de problemas, Creatividad). La plataforma ofrece a los profesores un interfaz web para personalizar (número de alumnos, herramientas, prácticas docentes, etc.) un Entorno Virtual Computacional (EVC), que es desplegado en la nube proporcionando un entorno integrado de todas herramientas necesarias para que el alumno genere un PDP y el profesor extraiga de forma automatizada las métricas para la evaluación de competencias específicas y transversales. La plataforma se encarga de aprovisionar los recursos de cómputo y almacenamiento además de la configuración para la puesta en marcha del EVC a través de proveedores Cloud (públicos o privados). Finalmente, se presenta un estudio de los tiempos de despliegue de EVCs sobre el proveedor cloud publico Amazon Web Services, demostrando que la plataforma abstrae de toda la complejidad de configuración e integración de las herramientas requeridas en un tiempo razonable.

Palabras clave: *la nube, competencias, programación software*

Abstract- This work presents a web platform to generate a Teaching Portfolio of software Programming (TPP) on the cloud, and the extraction of metrics of the generated TPP for assessing competences, both related to programming skills (e.g. Mastering the Object-Oriented Paradigm) as generic competences (e.g. Capacity of analysis and resolution of problems, Creativity). The platform provides the teachers with a web interface to customize (in terms of number of students, tools, lab sessions, etc...) a Computational Virtual Environment (CVE), which is deployed on the cloud providing an integrated environment composed of all required tools for the student to generate a TPP and the teacher to automatically extract the metrics for assessing both specific and soft skills. The platform is responsible for provisioning computational and storage resources, and also the configuration for starting up of the EVC on cloud providers (public or private). Finally, the paper presents study of the time required to deploy such CVEs in the Amazon Web Services public cloud provider, demonstrating that the platform abstracts the configuration complexity and integration of the required tools in a reasonable time.

Keywords: *cloud computing, competences, software programming*

1. INTRODUCCIÓN

Desde el año 1999, el Espacio Europeo de Educación Superior (EEES)¹ ha impulsado un modelo educativo fundamentado en un aprendizaje basado en competencias, con la puesta en marcha del plan Bolonia. Desde entonces, este organismo ha invertido un gran esfuerzo en identificar y estandarizar las competencias demandas por el mundo empresarial, tanto de forma general² como específica, por ejemplo, en el área de Tecnologías de la Información y la Comunicación (TIC)³.

En el ámbito de las titulaciones que contienen materias relacionadas con la programación de software, un método comúnmente utilizado para el desarrollo de competencias es la elaboración de Portafolios Docentes de Programación (PDPs), compuesto por códigos fuentes escritos por los estudiantes mediante un lenguaje (C, Java, Prolog, etc.) a través de las estructuras (selección, bucles, etc.), paradigmas (Orientado a Objetos, funcional, lógica, etc.) que estos ofrecen. Estos códigos fuentes son el resultado, entre otras cosas, de la resolución autónoma, o en grupo, de un conjunto de ejercicios o prácticas docentes de tamaño limitado, o también del resultado de desarrollar un proyecto software de mayor escala aplicando metodologías de desarrollo software (cascada, ágil, etc.). La información contenida en los PDPs resulta de mucha utilidad para extraer métricas (código duplicado, número de líneas de código, test unitarios, etc.), que resultan útiles para evaluar habilidades relacionadas con las competencias específicas asociadas a la programación. Por ejemplo, una métrica para evaluar la calidad del software puede ser el código duplicado. También es útil para evaluar si una competencia transversal ha sido adquirida por el estudiante o no. Por ejemplo, una métrica para evaluar la capacidad de análisis y resolución de problemas podría ser el número de test unitarios superados con éxito, o para evaluar la creatividad de los estudiantes se podría plantear la resolución de un problema con el mínimo número de líneas de código posible.

¹ European Higher Education Area and Bologna Process: <http://www.ehea.info>

² European Skills, Competences, Qualifications and Occupations (ESCO). <https://ec.europa.eu/esco/portal/home>

³ A common European framework for ICT Professionals in all industry sectors: <http://www.ecompetences.eu>

En la actualidad, en las ingenierías es habitual el uso de las TIC para el desarrollo de las competencias, a través de Actividades Educativas (A.E) diseñadas específicamente para ello. Esto se debe a que la combinación del uso de las TIC con estrategias o metodologías apropiadas mejoran los procesos de enseñanza aprendizaje de una forma significativa (Campbell, Bourne, Mosterman y Brodersen, 2002) (Fraser, Pillay, Tjatindi y Case, 2007). En el caso de los PDPs, las herramientas TIC no son solo necesarias para desarrollar los códigos fuentes sino también para extraer de estos las métricas que permitan una evaluación efectiva de las competencias de forma automática. Estas herramientas existen y las podemos enumerar, entre otras, en Sistemas de control de código fuente (p. ej. Git y la plataforma colaborativa on-line GitHub⁴, entornos de desarrollo integrado (p. ej. Code::Blocks⁵), herramientas de integración continua (p. ej. Jenkins⁶), frameworks de tests unitarios (p. ej. JUnit⁷ or CuTest⁸) y herramientas para evaluar la calidad de los códigos fuentes (p. ej. SonarQube⁹). Todas ellas son de uso común por parte de los desarrolladores en el mundo empresarial, por lo que su uso en la práctica docente, además de permitir la generación de PDPs y extracción de métricas, permite a los estudiantes conocer entornos realistas de trabajo y facilitar así la adquisición natural de las competencias demandadas por el mercado asociadas a la programación.

Combinar e integrar todas las herramientas comentadas en un único framework y de forma automatizada supone una significativa mejora para los docentes a la hora de poder utilizarlas y ponerlas en marcha en sus A.Es, dado que es un proceso que conlleva un coste temporal y una complejidad elevada, y que a día de hoy se realiza de forma manual. Además, muchas de estas herramientas software requieren de unos requisitos software y hardware que en muchas ocasiones no pueden ser satisfechas por los centros educativos. Por tanto, proporcionar una herramienta que además de combinar e integrar las herramientas suministre los recursos necesarios para su instalación, independientemente de los recursos del centro donde se realizan las A.Es, supone una mejora significativa para la comunidad docente.

Las tecnologías en la nube son suponen una oportunidad para mejorar el despliegue e integración de todas estas herramientas, mitigando los problemas comentados (González-Martínez, Bote-Lorenzo, Gómez-Sánchez y Cano-Parra, 2015) dado que estas tecnologías permiten el aprovisionamiento de recursos de cómputo, almacén y red creando Entornos Virtuales Computacionales (Sultan, 2010), (Xu, Huang y Tsai, 2014) configurados ad-hoc, utilizando un modelo de pago por uso de los recursos en el caso de utilizar proveedores de Cloud Público tales como Amazon Web Services (AWS)¹⁰. AWS es el líder de los proveedores cloud existentes en el mercado y ofrece una iniciativa, denominada AWS Educate¹¹, que proporciona a los estudiantes y

profesores becas que cubren el coste de la mayoría de recursos ofrecidos por AWS. Las universidades pueden unirse a este programa con el objeto de que sus estudiantes y profesores puedan aprovecharse de estas becas y sufragar los gastos del uso de los recursos AWS en el ámbito educativo, como es el caso del trabajo que se presenta en este artículo.

En la literatura también existen trabajos en otras áreas diferentes a la programación software, relativos al uso de las tecnologías en la nube para dar soporte a A.Es cuyo principal reto es reducir en laboratorios docentes informáticos el esfuerzo para crear, configurar y mantener los software requeridos para la ejecución de las actividades, además de proporcionar elasticidad y escalabilidad de los recursos en el caso de que hiciera falta. Por ejemplo, en (Xu, Huang y Tsai, 2014) se presentan laboratorios virtuales en la nube para la enseñanza de la seguridad en las redes. Además el trabajo presentado en (Salah, Hammoud y Zeadally, 2015), utiliza los recursos de AWS para proporcionar laboratorios virtuales para que los estudiantes puedan probar los conceptos de seguridad cibernética en un entorno seguro, reproducible y aislado. En el área de arquitectura, en un trabajo previo de los autores (Segrelles, Martínez, Castilla y Moltó, 2017) se desarrolló un EVC en la nube para desarrollar una experiencia trabajo en equipo a través de un aprendizaje basado en problemas.

2. CONTEXTO

El contexto de este trabajo se enmarca en el ámbito de todas aquellas titulaciones que tengan asignaturas que desarrollen la materia de programación software y apliquen PDPs como estrategia para su impartición.

El público objetivo será, por una parte los estudiantes que generarán los PDPs, por otra parte los técnicos de laboratorio o profesores que tienen que instalar y configurar todas las herramientas para generar PDPs y extraer las métricas, y finalmente los profesores que tienen que llevar a cabo las A.Es y evaluar las competencias adquiridas en base a las métricas extraídas de los PDPs generados.

A. Necesidad

La motivación de este trabajo surge por la necesidad de mitigar los costes temporales y complejidad de integración de aquellas herramientas que permitan la generación de PDPs y la extracción de métricas con el objeto de evaluar competencias específicas relacionadas con la programación de software y competencias transversales.

Estos problemas son debidos, por una parte, a los múltiples requisitos software de las diferentes herramientas a integrar, que pueden resultar incompatibles entre ellas o incluso con otras herramientas utilizadas en otras asignaturas donde se comparten los mismos recursos (p. ej. laboratorios). Por otra parte, los requisitos hardware suelen ser elevados, por lo que los equipos de los que se disponen pueden llegar a ser insuficientes para satisfacer la demanda del software siendo necesaria la realización de nuevas inversiones.

Estos problemas dificultan el uso de todas las herramientas de una forma integrada sin evitar la sobrecarga los equipos técnicos de los laboratorios o la inversión en nuevos recursos computacionales, que en muchos de los casos es inasumible. Además, estos problemas pueden agravarse en entornos puramente on-line, donde el número de estudiantes puede ser elevado y muy variable, y por lo tanto los recursos hardware y

⁴GitHub Repository: <https://github.com/>

⁵Code::Blocks: <http://www.codeblocks.org/>

⁶Jenkins: <https://jenkins.io>

⁷JUnit: <http://junit.org/junit4>

⁸CuTest: <http://cutest.sourceforge.net/>

⁹SonarQube: <https://www.sonarqube.org>

¹⁰AWS: <https://aws.amazon.com/>

¹¹AWS Educate:

<https://aws.amazon.com/es/education/awseducate/>

software deben de proporcionarse y ajustarse dinámicamente, necesitando escalar el número de ellos en función de la demanda.

B. Objetivos

El objetivo de este trabajo es desarrollar una plataforma web que despliegue de forma automatizada un EVC en la nube configurados para que los alumnos generen PDPs y los profesores extraigan métricas que permitan evaluar competencias específicas asociadas la programación de software y competencias transversales. La plataforma abstraerá al profesor o técnicos de laboratorios de la complejidad de configuración e integración de las diferentes herramientas software requerido y proporcionará los recursos en la nube de forma escalable y flexible en función del número de alumnos y herramientas utilizadas. La plataforma permitirá desplegar EVC tanto en proveedores cloud públicos (p. ej. AWS) como privados (p. ej. OpenNebula¹²).

3. DESCRIPCIÓN

El EVC a desplegar desde la plataforma Web se compone de un conjunto de herramientas software (ver Figura 1) que son integradas en un único entorno de trabajo desplegado en la nube mediante Máquinas Virtuales (MV). Todas estas herramientas software permitirán generar PDPs y extraer automáticamente métricas como las mostradas en la Tabla 1.

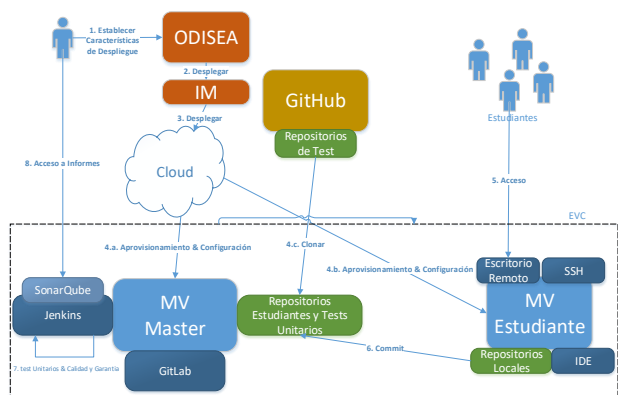


Figura 1. Arquitectura de la plataforma para desplegar EVC junto con la estructura y flujo de las actividades.

A. Herramientas Software

Las herramientas software que componen el EVC son:

- **Code::Blocks.** Entornos de Desarrollo Integrado (IDE) de libre distribución y multiplataforma para desarrollar programas en los lenguajes C y C++. Esta herramienta ha sido diseñado para poder ser extendida a otros lenguajes (p.ej. Java). En el EVC, se instala en una MV donde los estudiantes se conectan y generan los códigos fuentes que conformarán el PDP (MV Estudiante en la Figura 1).
- **GitLab.** Plataforma para desarrollar proyectos software utilizando un control de versiones distribuido y la funcionalidad de gestión de códigos fuentes de Git¹³. En

el EVC, la herramienta cliente (git) se instala en las MV estudiante y la herramienta servidor se instala en un servidor dedicado en una MV Maestra (ver Figura 1).

- **Jenkins.** Herramienta de código abierto de integración continua de código fuente para medir automáticamente la calidad y garantía del software a través de diferentes tipos de test (p.ej. test unitarios). Los códigos de los test deben ser previamente diseñados por el profesor y guardados en un repositorio GitHub. En el EVC, Jenkins se instala en una MV Maestra (ver Figura 1). Jenkins y GitLab pueden instalarse en una misma MV maestra o en dos MV maestras independientes, dependiendo de los parámetros configurados por el profesor en la plataforma web.
- **SonarQube.** Plataforma de código abierto para inspeccionar la calidad de código. Se utiliza para extraer de forma automática métricas relativas a la calidad de los códigos fuentes de los PDP generados por los alumnos. Esta herramienta es instalada como un plugin en la herramienta Jenkins, por lo que también es instalado y configurado en una MV Maestra (ver Figura 1).
- **JUnit.** Es un framework para test unitarios de programas escritos en Java y es utilizado por el EVC para realizar test unitarios de los PDPs que utilizan Java como lenguaje de programación. Los test unitarios son clonados de un repositorio en GitHub (ver Figura 1).

Tabla 1. Métricas de las herramientas software integradas en el EVC.

Sw	Métricas	Descripción
GitHub	Contribuciones	Contribuciones (<i>commits</i> , agregación o borrado de código) en el tiempo.
	<i>Commits</i>	Número de <i>commits</i> por día/semana.
	Frecuencia de Código	Agregaciones o borrados de código por semana.
	Punch Card	Número de <i>commits</i> por hora.

Sonar Qube	Líneas de Código	Número de líneas de código (no comentarios) con al menos un carácter.
	Líneas de Comentarios	Número de líneas de código de tipo comentario.
	Comentarios (%)	Densidad de comentarios = (Líneas de comentarios) / (Líneas de código + Líneas de Comentarios)*100
	Bloques Duplicados	Número de bloques duplicados de líneas de código.

Jenkins	Tests exitosos	Número de test unitarios que se han superado con éxito.
	Tests Fallidos	Número de test unitarios que no se han superado con éxito.

- **CUTest.** Es un framework para test unitarios de programas escritos en C y es utilizado por el EVC para realizar test unitarios de PDPs que utilizan C como lenguaje de programación. Los test unitarios son clonados de un repositorio de GitHub (ver Figura 1).

La arquitectura reflejada en la Figura 1 está diseñada para la creación de EVC que tengan un tiempo de vida igual al de las A.E. Sin embargo, los test unitarios se almacenan en GitHub, por lo que pueden reaprovecharse los test unitarios en los nuevos EVC que se creen en A.E futuras.

¹² OpenNebula - Flexible Enterprise Cloud Made Simple: <https://opennebula.org>

¹³ Git: <https://git-scm.com>

B. ODISEA

El principal componente de la plataforma web desarrollada en este trabajo es ODISEA (*On-demand Deployment of Infrastructures to Support Educational Activities* (Segrelles, Moltó, Caballer, 2015). (Segrelles y Moltó, 2016), que permite desplegar EVC en la nube especialmente diseñados para la ejecución de A.E. En trabajos anteriores, el impacto positivo y el potencial de ODISEA se analizó desde el punto de vista de los profesores (Segrelles et al., 2015), de los centros de educación (Segrelles y Moltó., 2016) y de los estudiantes (Segrelles y Moltó, 2016). ODISEA, permite a través de recetas, donde se describen los requerimientos software, hardware y configuración, desplegar de forma automática EVCs en diferentes proveedores cloud.

C. Flujo de Actividades para el despliegue de EVCs en la nube

En este trabajo se ha desarrollado una plataforma web que permite a los profesores configurar las características o parámetros de un EVC a través de un interfaz web amigable. A partir de estos parámetros la plataforma genera automáticamente un conjunto de recetas que son interpretadas por ODISEA, con el objeto de que este despliegue en la nube (pública o privada) el EVC completamente configurado, integrando todas las herramientas comentadas en la sección A y proporcionando los recursos de cómputo y almacenamiento requeridos.

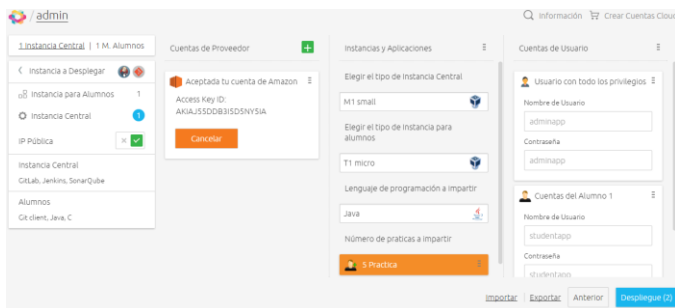


Figura 2. Pantalla para la configuración de los parámetros de despliegue del EVC (proveedor Cloud, Número de MVs de estudiantes, etc.).

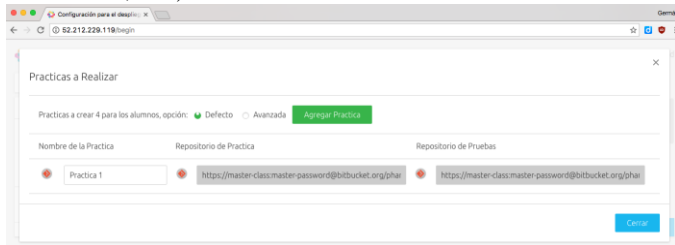


Figura 3. Pantalla para la configuración de los parámetros de despliegue del EVC relativos a los repositorios del estudiante (test unitarios y plantillas de códigos fuentes).

La Figura 1 muestra el flujo de las diferentes actividades que se van ejecutando cuando se despliega un EVC a través de la plataforma Web. Primero, el profesor accede a la plataforma web y configura los parámetros de despliegue (ver Figura 2 y Figura 3), a continuación la plataforma genera las recetas y se las envía a ODISEA para proceder a la configuración y despliegue en la nube del VCE (paso 1 - Figura 1). Para el despliegue, ODISEA utiliza el IM (*Infrastructure Manager*)

(Fernández, Espert, Moltó, Laguna, 2015) que es el componente encargado de aprovisionar los recursos desde un proveedor Cloud y configurarlos automáticamente para crear VCEs, que en nuestro caso será la creación un VCE que integre todas las herramientas mencionadas en el subapartado A (pasos 2 y 3 - Figura 1).

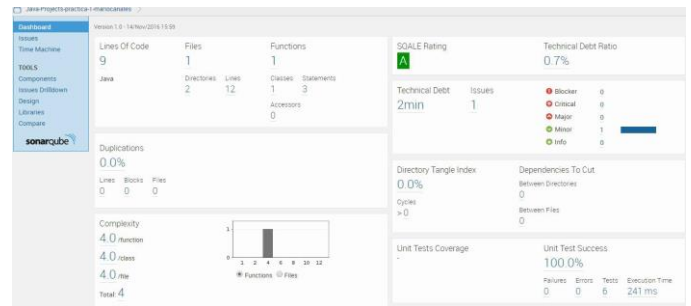


Figura 4. Informes de SonarQube sobre la calidad del código fuente.

En el proceso de despliegue del VCE, en primer lugar, la MV maestra es desplegada y configurada para dar soporte a las herramientas GitLab, Jenkins y SonarQube. Los repositorios de cada estudiante requeridos para las A.Es y los test unitarios integrados en Jenkins son creados automáticamente (paso 4.a - Figura 1). En segundo lugar, las MVs de los estudiantes se despliegan (paso 4.b - Figura 1). Estas pueden ser muchas en función de los parámetros iniciales de configuración indicados en el paso 1. Cada MV de estudiante es equipada con un IDE, que puede ser accedido por escritorio remoto, y una copia local de los repositorios Git que necesita el estudiante para realizar las A.Es (paso 4.c. - Figura 1). Cabe destacar que los estudiantes pueden acceder a su MV mediante SSH o mediante RDP (Escritorio Remoto) desde sus propios ordenados (paso 5 - Figura 1), fomentando así el BYOD (*Bring Your Own Device*). En tercer lugar, los estudiantes llevan a cabo la A.E planeadas, generando los códigos fuentes de los PDPs en los repositorios de códigos locales, y de forma periódica confirmando los cambios en los repositorios de GitLab alojado en la MV maestra. Cuando se confirma un cambio en los clientes, se activan los test unitarios configurados en Jenkins automáticamente con el fin de realizar los test unitarios y extraer las métricas correspondientes en SonarQube (pasos 6 y 7 - Figura 1). Cuando la A.E. finaliza, el profesor accede a los informes a través del interfaz de usuario gráfico proporcionado por Jenkins y sonarQube (ver Figura 4).

Cabe destacar que toda la complejidad de desplegar y personalizar los programas se evita al profesor, dado que todas las actividades descritas a excepción del paso 1 se realizan automáticamente. El principal beneficio para los estudiantes es que estos reciben el mismo entorno de trabajo que los utilizados en el mundo laboral, independientemente de la capacidad sus ordenadores, dado que el propio EVC reside en la nube en lugar de en sus propios dispositivos.

4. RESULTADOS

En esta sección se presenta un análisis de los tiempos de despliegue de EVCs sobre el proveedor público AWS utilizando la plataforma web. Para ello, se ha elegido la región

de AWS *us-east-1* y utilizado las instancias *m1.small* (1.7 GiB RAM, 1 vCPU, 160 GiB disco duro). Las características introducidas a través de la plataforma web para el despliegue de los VCEs son las siguientes:

- Una única instancia de la MV maestra que contiene GitLab y Jenkins con el Plugin de SonarQube instalado.
- 10 cuentas de estudiantes.
- 5 programas software a desarrollar (lecciones de prácticas) por el estudiante. Esto implica 5 repositorios GitLab por estudiante, resultando en un total de 50 repositorios.

El escenario de prueba también tiene en cuenta un repositorio GitHub que contiene los test unitarios para las diferentes lecciones prácticas, sin embargo estas no afectan al tiempo de despliegue, dado que están en GitHub y no en el EVC.

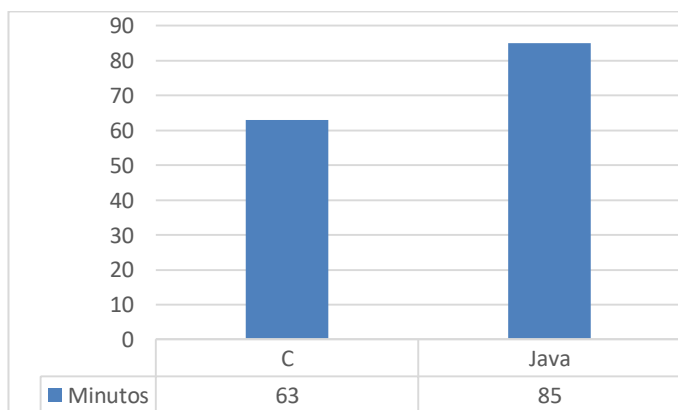


Figura 5. Tiempo de despliegue de la MV Maestra en AWS.

Los tiempos de despliegue y configuración de la MV maestra se muestran en la Figura 5. Los tiempos resultantes son elevados (del orden de una hora). Esto es atribuible principalmente a dos razones, que son las siguientes:

- El uso de instancias en AWS de tipo *m1.small*, que ofrece recursos de cómputo escasos. Esto minimiza el coste, dado que AWS factura por hora de ejecución. Sin embargo, la falta de memoria y la potencia de cálculo incrementa significativamente el tiempo de instalación de las herramientas.
- El despliegue se basa en imágenes base de MV (Ubuntu 14.04) en las que todo el software se instala dinámicamente y se configuran en el momento del despliegue. Esto introduce una penalización en el tiempo de despliegue en comparación con el uso de imágenes de MV preinstaladas. Sin embargo, el uso de recetas automáticas para la configuración permite a ODISEA desplegar el mismo EVC en diferentes proveedores Cloud.

Destacar, que los tiempos para desplegar un EVC para evaluar PDPs generados en C son significativamente más bajos que los PDPs basados en Java. Esto se debe a que este último caso implica la instalación de SonarQube y los complementos de integración correspondientes para Jenkins, lo cual supone un mayor número de herramientas a instalar y configurar.

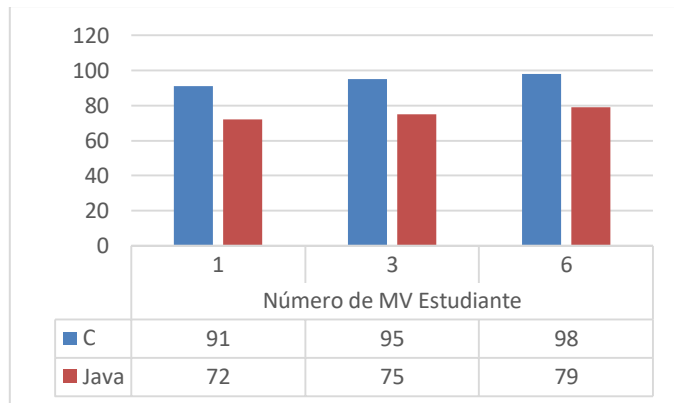


Figura 6. Tiempo de despliegue del EVC.

La Figura 6 muestra el tiempo de despliegue de los EVCs, incluyendo las MV de los estudiantes, considerando diferentes números de estudiantes en la configuración. Destacar que los tiempos para 1, 3 y 6 instancias son similares dado que las instancias de los estudiantes son desplegadas y configuradas en paralelo, por lo que número de estudiantes no afecta al tiempo global de forma significativa. También destacar que los requisitos de configuración de las MV de los estudiantes son mucho más ligeros que las MV maestras, por lo que se despliegan en un tiempo reducido después de desplegar las MV maestras.

5. CONCLUSIONES

El trabajo presenta una plataforma web que permite desplegar EVCs en la nube para la generación de PDPs y la evaluación de competencias específicas relativas a la programación software y también la evaluación de competencias transversales. Dichas evaluaciones se realizan a partir de la extracción automática de métricas estándar de los códigos fuentes de los PDPs utilizando una serie de herramientas software integradas. La plataforma proporciona al profesor con un interfaz web que permite personalizar el EVC y abstraer de la complejidad de instalación de las herramientas software requeridas y su integración.

La capacidad de desplegar automáticamente el EVC en una nube presenta ventajas significativas en comparación con una configuración de laboratorio tradicional, incluso cuando se emplea la virtualización para preestablecer una imagen de una MV en los equipos de un laboratorio. El uso de recetas de configuración utilizado por ODISEA permite desplegar dinámicamente software en imágenes de MV. Este es un proceso determinista que genera un mismo entorno disponible para los estudiantes después de cada despliegue. Además, la automatización del proceso evita errores típicos que ocurren en las configuraciones manuales.

La plataforma permite utilizar proveedores de recursos en la nube tanto privados como públicos, posibilitando escalar los EVC de forma elástica dependiendo del número de estudiantes que van a desarrollar la A.E, y por tanto siendo una herramienta idónea para entornos on-line.

La plataforma puede ser utilizada por cualquier asignatura que utilice PDPs escritos en C o Java, siendo fácilmente extensible a otros lenguajes de programación, extendiendo las recetas generadas automáticamente para que instale el software requerido.

La plataforma es de código abierto y ha sido liberado bajo licencia Apache 2.0 y es públicamente disponible en GitHub¹⁴.

Xu, L., Huang, D. y Tsai, W. T. (2014). Cloud-based virtual laboratory for network security education. *IEEE Transactions on Education*, 57(3), 145-150.

AGRADECIMIENTOS

En este trabajo, los autores agradecen por la financiación recibida del Vicerrectorado de Estudios, Calidad y Acreditación de la Universitat Politècnica de València (UPV) para desarrollar el Proyecto de Innovación y Mejora Educativa (PIME) “Entornos Virtuales Computacionales para la Evaluación de Competencias Transversales en la Nube”, con referencia A04.

REFERENCIAS

- Campbell, J. O., Bourne, J. R., Mosterman, P. J. y Brodersen, A. J. (2002). The effectiveness of learning simulations for electronic laboratories. *Journal of Engineering Education*, 91(1), 81-87.
- Fernández, M. C., Espert, I. B., Moltó, G. y Laguna, C. D. A. (2015). Dynamic management of virtual infrastructures. *In Journal of Grid Computing*, Vol. 13, No. 1, pp. 53-70. Springer Verlag (Germany).
- Fraser, D. M., Pillay, R., Tjatindi, L. y Case, J. M. (2007). Enhancing the learning of fluid mechanics using computer simulations. *Journal of Engineering Education*, 96(4), 381.
- González-Martínez, J. A., Bote-Lorenzo, M. L., Gómez-Sánchez, E. y Cano-Parra, R. (2015). Cloud computing and education: A state-of-the-art survey. *Computers & Education*, 80, 132-151.
- Salah, K., Hammoud, M. y Zeadally, S. (2015). Teaching Cybersecurity Using the Cloud. *IEEE Transactions on Learning Technologies*, 8(4), 383-392.
- Segrelles, J. D., Martínez, A., Castilla, N. y Moltó, G. (2017). Virtualized Computational Environments on the cloud to foster group skills through PBL: A case study in architecture. *Computers & Education*, 108, 131-144.
- Segrelles, J. D. y Moltó, G. (2016). Assessment of cloud-based Computational Environments for higher education. *In Frontiers in Education Conference (FIE)*, 2016 IEEE (pp. 1-9). IEEE.
- Segrelles, J. D., Moltó, G. y Caballer, M. (2015). Remote Computational Labs for Educational Activities via a Cloud Computing Platform. *In 2015 Proceedings of the Information Systems Education Conference (ISECON)* (pp. 309-321).
- Sultan, N. (2010). Cloud computing for education: A new dawn?. *International Journal of Information Management*, 30(2), 109-116.
- Xu, L., Huang, D. y Tsai, W. T. (2014). Cloud-based virtual laboratory for network security education. *IEEE Transactions on Education*, 57(3), 145-150.

¹⁴ Repositorio en GitHub de ODISEA:
<https://github.com/grycap/odisea>