

Tipo de artículo: Artículo original

Computación en la nube para sistemas distribuidos

Cloud computing for distributed systems

Bolívar Ramos Mosquera^{1*} , <https://orcid.org/0000-0001-9629-06871>

¹ Universidad de Guayaquil. Ecuador. Correo electrónico: bolivar.ramosm@ug.edu.ec

* Autor para correspondencia: bolivar.ramosm@ug.edu.ec

Resumen

El desarrollo de sistemas de software es un tema crucial y significativamente desafiante, en el plan de estudios de Ingeniería en sistemas. La asignatura Sistemas Distribuidos ha sido reconocida de naturaleza abstracta y difusa, con varias definiciones, principios, conceptos y nociones teóricas. En este sentido, los autores coinciden en la necesidad de permitir que los estudiantes comiencen practicando el diseño arquitectónico y la implementación de sistemas de software ligeros, para aplicar los elementos teóricos del programa. La nube proporciona un entorno para crear y mantener productos y servicios basados en software. Este trabajo tiene como objetivo el diseño de una estrategia educativa para desarrollar proyectos ligeros de sistemas distribuidos con recursos de la nube públicos, de manera que los estudiantes adquieran experiencia en el diseño arquitectónico e implementación de sistemas distribuidos, y fortalezcan los conocimientos básicos de ingeniería de sistemas. La estrategia propuesta cuenta con seis actividades básicas: (1) Diagnóstico; (2) Planificación; (3) Capacitación; (4) Implementación; (5) Evaluación; y (6) Reutilización. Se diseñó para ser implementada en un periodo de seis meses, y está dirigida a estudiantes del III y IV periodo de la carrera de carrera Tecnologías de la Información. Los resultados de la investigación muestran que la implementación de proyectos ligeros con recursos de la nube pública, tiene gran utilidad para la comprensión de conceptos básicos de sistemas distribuidos y diseño arquitectónico. Para el proceso de enseñanza se recomienda IMB Cloud Lite por ser gratis y contener servicios que son suficientes para este tipo de objetivos.

Palabras clave: sistemas distribuidos; diseño arquitectónico; PaaS; computación en la nube; estrategia educativa

Abstract

Software systems development is a crucial and significantly challenging topic in the Systems Engineering curriculum. The Distributed Systems subject has been recognized as abstract and diffuse in nature, with various definitions, principles, concepts and theoretical notions. In this sense, the authors agree on the need to allow students to start practicing architectural design and the implementation of lightweight software systems, in order to apply the theoretical elements of the program. The cloud provides an environment to create and maintain software-based products and services. This work aims to design an educational strategy to develop light projects of distributed systems with public cloud resources, so that students gain experience in the architectural design and implementation of distributed systems, and strengthen the basic knowledge of systems engineering. The proposed strategy has six basic activities: (1) Diagnosis; (2) Planning; (3) Training; (4) Implementation; (5) Evaluation; and (6) Reuse. It was designed to be implemented in a period of six months, and is aimed at students of the III and IV period of the Information Technology career. The research results show that the implementation of light projects with public cloud resources is very useful for understanding basic concepts of distributed systems and architectural design. For the teaching process, IMB Cloud Lite is recommended because it is free and contains services that are sufficient for this type of objective.

Keywords: distributed systems; architectural design; PaaS; Cloud Computing; educational strategy

Recibido: 15/04/2023

Aceptado: 26/07/2023

En línea: 01/08/2023



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

Introducción

El desarrollo de sistemas de software es un tema crucial y significativamente desafiante en el plan de estudios de Ingeniería en sistemas. Para abordar la naturaleza compleja y confusa de este tema, los educadores todavía enfrentan varias dificultades en la práctica, como los recursos de implementación para los proyectos de los estudiantes; los factores de administración y gestión tecnológica; las preocupaciones financieras y legales; y la plétora de información sobre los ecosistemas de desarrollo que dificulta la selección del marco de trabajo.

La necesidad de definir un entorno tecnológico robusto es una prioridad de todas las organizaciones dedicadas al desarrollo de productos y servicios de software, así como de las Instituciones de la Educación Superior (IES), que tienen la responsabilidad de formar profesionales en el campo de la ingeniería de software (Sánchez & Barrezueta, 2022). En consecuencia, desde la perspectiva de los autores, todos los estudiantes de la carrera Ingeniería en sistemas, deben estar equipados con el conocimiento tanto teórico como práctico del diseño e implementación de software.

La asignatura Sistemas Distribuidos ha sido reconocida de naturaleza abstracta y difusa, con varias definiciones, principios, conceptos y nociones teóricas (Schümmer et al., 2017). Intentar ofrecer un conocimiento completo de la arquitectura distribuida no solo requerirá largas sesiones de enseñanza, sino que también introducirá confusiones inesperadas en la comprensión de los estudiantes. Estos desafíos pueden y deben abordarse gradualmente, en lugar de todos a la vez (Cico et al., 2021). En este sentido, los autores coinciden en la necesidad de permitir que los estudiantes comiencen practicando el diseño arquitectónico y la implementación de sistemas de software ligeros, para aplicar los elementos teóricos del programa. Esta integración permite a los estudiantes aplicar conocimientos básicos de ingeniería de software, a la vez que se preparan para el desarrollo de sistemas más complejos.

En este contexto, la Computación en la Nube (*Cloud Computing*, CC) proporciona un entorno altamente flexible que ofrece una variedad de servicios. La CC es un paradigma de computación distribuida que permite el acceso a recursos virtualizados que incluyen computadoras, redes, almacenamiento, plataformas de desarrollo o aplicaciones. Los proveedores de servicios de la nube son empresas que ofrecen entornos de TI los cuales pueden prestar servicios en línea que incluyen: la Infraestructura como servicio (IaaS), Plataforma como servicio (PaaS) y Software como servicio (SaaS) (Arunarani et al., 2019).

PaaS es un modelo de servicio en la nube donde los proveedores de servicios entregan una plataforma al cliente para desarrollar y hospedar la aplicación. Los desarrolladores utilizan esta plataforma para desarrollar, probar y ejecutar la aplicación. Los desarrolladores pueden concentrarse en crear sus aplicaciones y dejar las preocupaciones de configurar la plataforma con los proveedores de servicios de la plataforma. En la revisión de la literatura se destacan como las



PaaS más empleadas Google App Engine (Ferrer et al., 2016); Virtuozzo (Tsuruoka, 2016); Azure App Service (Ferrer et al., 2016); AWS Elastic Beanstalk (Arulkumar & Lathamaju, 2019); e IBM Cloud (Ray, 2016).

En el estudio realizado por (Li, 2020) se identificaron como principales beneficios del uso PaaS para la enseñanza, el desarrollo rápido, la integración continua, el ahorro de costos, el uso compartido de código, y el uso de un entorno de desarrollo ya configurado. Los riesgos clave determinados son las dependencias, la falta de disponibilidad de acceso a la nube, el compromiso y la integración del código, la deuda técnica y los costos de soporte adicionales. Los resultados revelaron que si dichos entornos no se planifican y configuran con cuidado, los beneficios de usar la nube para el proceso de Enseñanza Aprendizaje de las IES, podrían verse eclipsados por los riesgos asociados con ella.

De acuerdo con la revisión realizada por los autores, en la literatura hay pocos informes disponibles sobre la enseñanza de la asignatura Sistemas Distribuidos utilizando la nube como entorno de desarrollo principal. Si bien se ha investigado en la literatura el potencial de usar la nube para la Educación, aún es limitada la información sobre la arquitectura de promulgación de procesos de software basada en la nube que utiliza la elasticidad, accesibilidad y disponibilidad de la nube para facilitar el desarrollo distribuido, y superar algunos de los desafíos técnicos y de comunicación asociados (González-Martínez et al., 2015). Para poner en práctica estos elementos, se analizan las bondades de los recursos gratuitos y disponibles públicamente de la Plataforma como servicio (PaaS) del mercado de la nube pública.

La PaaS pública ofrece cuotas gratuitas que son suficientemente para que los estudiantes implementen proyectos pequeños. De hecho, al centrar el estudio en escenarios de software distribuido, resulta conveniente y flexible enseñar conceptos y patrones arquitectónicos con los recursos PaaS disponibles públicamente. Para esta investigación se propone utilizar IBM Cloud Lite. Este tipo de cuenta no tiene costo y te permite utilizar la mayoría de los servicios de la nube. Lite se refiere a servicios que tienen una opción de prueba gratuita, por eso es esencialmente útil en el proceso de enseñanza (Song et al., 2022).

Este trabajo tiene como objetivo el diseño de una estrategia educativa para desarrollar proyectos ligeros de sistemas distribuidos con recursos PaaS públicos, de manera que los estudiantes adquieran experiencia en el diseño arquitectónico e implementación de sistemas distribuidos, y fortalezcan los conocimientos básicos de ingeniería de sistemas. En esta investigación se propone el uso de PaaS para la creación de proyectos ligeros de los estudiantes de la carrera Ingeniería en sistemas computacionales, de la Universidad de Guayaquil en Ecuador.

Materiales y métodos

Se propone una estrategia educativa para utilizar la computación en la nube como entorno de desarrollo de sistemas distribuidos, donde los estudiantes implementen proyectos pequeños, que permita alcanzar experiencia en el diseño de arquitectura de software e implementación de sistemas distribuidos. La propuesta se realiza sobre el servicio Pass de la



nube pública, y se validan los resultados del estudio basado en una investigación de acción. La estrategia cuenta con seis actividades básicas: (1) Diagnóstico; (2) Planificación; (3) Capacitación; (4) Implementación; (5) Evaluación; y (6) Reutilización. Se diseñó para ser implementada en un periodo de seis meses, y está dirigida a estudiantes del III y IV periodo de la carrera de carrera Tecnologías de la Información.

La Figura 1 muestra las actividades básicas de la estrategia educativa propuesta, y la secuencia y relación de cada una de estas. Fue diseñada para ejecutarse por iteraciones a partir de los resultados de la observación, la evaluación, la acumulación de experiencias y los objetivos de aprendizajes especificados para cada caso.

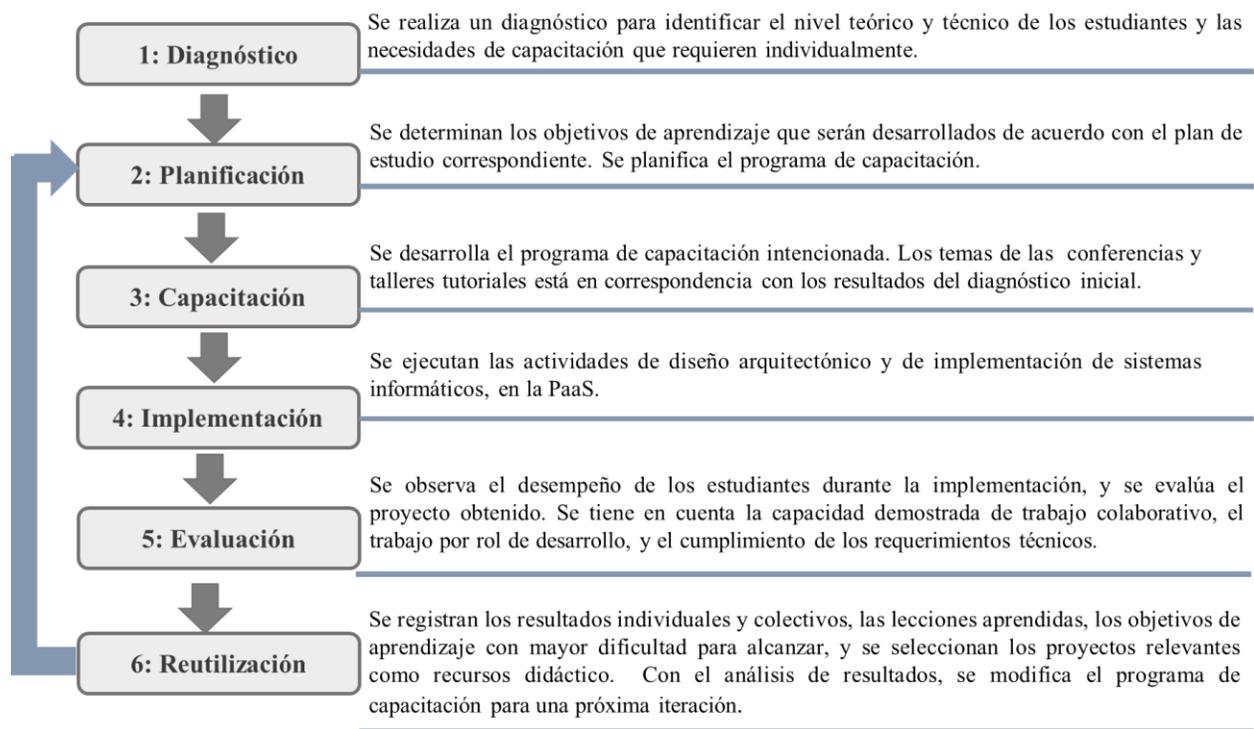


Figura 1. Actividades básicas de la estrategia educativa.

Fuente: Elaboración propia.

Para recopilar evidencia empírica sobre el impacto de la estrategia educativa propuesta, se empleó el Modelo de Aceptación de Tecnología (TAM) (Davis, 1989) para validar qué tan bien los estudiantes recibieron dicha estrategia educativa. La investigación se realizó sobre una muestra de estudiantes del III y IV periodo de la carrera Ingeniería en sistemas, durante seis meses.

Resultados y discusión



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional (CC BY 4.0)**

La estrategia fue implementada sobre una muestra de 20 estudiantes del III y IV periodo de la carrera Tecnologías de la Información, en un periodo de seis meses. Las actividades fueron ejecutadas como sigue:

1 Diagnóstico

Se realizó un diagnóstico inicial para conocer el nivel de conocimientos de los estudiantes sobre arquitectura de software, patrones arquitectónicos, sistemas distribuidos, Entornos Integrado de Desarrollo, trabajo colaborativo, metodologías ágiles de desarrollo, lenguajes de programación, gestión de requisitos. En esta actividad los estudiantes respondieron elementos conceptuales y realizaron el modelado de un proyecto pequeño en UML (*Unified Modelling Language*), asociado a la gestión de la información del personal docente de la universidad, y la gestión de los resultados docentes de los estudiantes.

Para realizar el diagnóstico se planearon proyectos individuales con aplicaciones simples, y se dio una semana para entregar el modelado. Los resultados demostraron que, el 100% tiene capacidad de trabajo colaborativo y el 75% tiene un buen dominio de los procesos de gestión de requisitos; sin embargo, en el 87% de los estudiantes se identificaron deficiencias en el diseño arquitectónico. Asimismo manifiestan que es difícil comprender los elementos básicos de la arquitectura distribuida sin explorar sobre una implementación real.

2 Planificación: Basado en la maya curricular de las asignaturas sistemas distribuidos y arquitectura de software, y teniendo en cuenta los resultados del diagnóstico inicial realizado, se definieron los objetivos de aprendizaje que se pretenden fortalecer con la implementación de la estrategia educativa. Se realizó además la planificación de la capacitación introductoria que recibirán los estudiantes, y del proceso de evaluación de la estrategia.

Inicialmente se decidió ejecutar 2 iteraciones de la estrategia, por lo que se planificaron los temas genéricos de la primera iteración de capacitación, ya que la segunda iteración se realiza en función de los resultados y hallazgos registrados en la actividad de Reutilización y observación. Cada iteración tendrá 3 meses para ser ejecutada integralmente. En cada una se impartirá una ronda de conferencias, y una ronda de asimilación del entorno tecnológico en clases de laboratorio.

3 Capacitación:

Tres profesores se encargaron de la capacitación por conferencias, para abordar los temas que mayores dificultades se identificaron en el diagnóstico inicial; de manera general se abordó la estimación del rendimiento de un sistema de software distribuido; patrones de interconexión de componentes; encapsulamiento de los componentes del cliente y del servidor; las propiedades visibles externamente de los componentes de software y las interconexiones entre ellos, la especificación formal de las interacciones usando un Lenguaje de Descripción de Arquitectura (ADL). Dado que la computación distribuida está ganando cada vez más popularidad e importancia en la industria del software y la



educación, es especialmente importante la enseñanza de un conjunto de patrones arquitectónicos distribuidos, incluido el patrón Cliente-Servidor (Glushan et al., 2015), el patrón Modelo-Vista-Controlador (MVC) (Lee & Wang, 2019), el patrón Microservices -Architecture (MSA) (Li et al., 2021), entre otros que también son explicados.

Para la ronda de capacitación de laboratorio fue necesario crear un manual de usuario paso a paso para el trabajo con PaaS. Dado que se definió IBM Cloud para la implementación, la preparación se enfocó solo en esta plataforma. El objetivo de esta preparación es lograr que los estudiantes se familiaricen con el entorno de desarrollo en la nube, para que en la actividad de implementación se centren en los aspectos técnicos del desarrollo de software distribuidos, en lugar de sentirse abrumados, y con una carga extra en la asimilación tecnológica.

Los materiales didácticos y los tutoriales fueron adaptados para diferentes proyectos e incluso para diferentes estudiantes, de acuerdo con sus diferencias individuales. Los recursos didácticos sobre el servicio PaaS fueron enriquecidos atendiendo los objetivos de aprendizaje que se persiguen con la implementación de la propuesta. En la primera iteración se impartieron dos laboratorios para que los estudiantes se familiarizaran con los servicios PaaS disponibles, y de esta forma aumentar la facilidad de uso de la plataforma y verificar la utilidad de IBM Cloud. El primer laboratorio constituyó una introducción al ecosistema de desarrollo, para comenzar a entender la plataforma y los servicios que ofrece. El segundo laboratorio se enfocó en el desarrollo de métodos simples utilizando *Cloud Functions*. A modo de ejemplo se definió *Liberty for Java*, pero puede elegir cualquier opción.

4 Implementación:

En esta actividad se determina la metodología de desarrollo de software a utilizar. Al ser un proyecto pequeño, con un equipo de desarrollo igualmente pequeño, es lógico definir una metodología ágil de desarrollo. La computación en la nube tiene una gran combinación con el desarrollo ágil de software como área de investigación. Muchos investigadores trabajaron en el campo *Agile Cloud*; para este caso se propone DevOps (Schwaber & Beedle, 2002).

Se definen los requisitos funcionales que debe cumplir el proyecto, con respecto a un patrón arquitectónico específico. Se seleccionan los recursos tecnológicos a emplear, y finalmente se implementa el proyecto. El módulo de cliente se mantuvo localmente, y la implemente del módulo de servidor en IBM Cloud. Se definió Lite porque ofrece un tipo de cuenta gratuita y sin vencimiento con 256 MB de memoria de tiempo de ejecución de *Foundry* en la nube instantánea, además, da acceso con límite de uso a servicios seleccionados como Internet of Things Platform y Data Science Experience. La base de datos fue Db2, es una base de datos SQL relacional que usa XML. *Cloud Functions* se utilizó para desarrollar el *back-end* y por defecto este viene con Node.js, el cual es un entorno de ejecución que utiliza JavaScript. Los estudiantes terminaron sus proyectos en un promedio de 20 días, periodo en el que colaboraron activamente.



5 Evaluación: En esta actividad se ejecutan las acciones docentes de asignación y supervisión del proyecto, en lugar de las actividades estudiantiles de implementación del proyecto. Puede realizarse en paralelo una vez iniciada la actividad Implementación. Como la evaluación se realiza basado en los proyectos implementados por los estudiantes, en lugar de evaluar todo el periodo, se recomienda realizar varias iteraciones dentro de un semestre. Para efectos de esta investigación, se ejecutaron 2 iteraciones completas en un periodo de 6 meses. Desde la perspectiva de los profesores las plataformas PaaS de acceso público facilitan la evaluación de los proyectos de los estudiantes al permitir comprobar en línea el tiempo de ejecución de los sistemas (o componentes) del proyecto. Por el contrario, antes de emplear esta estrategia educativa, la verificación de los informes de proyectos normalmente requería que los profesores y asistentes de enseñanza observaran las demostraciones en vivo de los estudiantes en el laboratorio o incluso que compilaran sus códigos fuente localmente.

Las evaluaciones se realizan cualitativa y cuantitativamente, de manera que el informe cualitativo, se emplee en el perfeccionamiento de la actividad de Planificación.

6 Reutilización: Los resultados obtenidos por cada estudiante, fundamentalmente el sistema implementado, y el diseño de la arquitectura base, se pueden reutilizar directamente como material didáctico para temas ingenieriles de interés en otros periodos y asignaturas como calidad del software, integración continua, escalabilidad, rendimiento, tolerancia a fallos, entre otros. La reutilización permite compartir proyectos de estudiantes que resultaron relevantes entre estudiantes de periodos diferentes a la muestra objeto de estudio, constituyendo así recursos educativos para la impartición de la asignatura Sistemas Distribuidos.

Evaluación de la propuesta

En esta sección se expone la experiencia y las lecciones aprendidas de la implementación de dos iteraciones de la estrategia educativa para creación de sistemas distribuidos en la PaaS de la nube pública. Se ha empleado TAM para ayudar a reflejar el impacto que tuvo la estrategia educativa propuesta para los estudiantes en términos de:

- Utilidad percibida: indica en qué medida los estudiantes creen que han avanzado en la capacidad de comprender los elementos arquitectónicos un sistema de software distribuido.
- Facilidad de uso percibida: indica en qué medida los estudiantes creen que han implementado en la PaaS el proyecto de software sin mucha dificultad o gran esfuerzo.

La encuesta identifica y analiza las ventajas y dificultades que percibieron los estudiantes a partir de la implementación de proyectos ligeros en la PaaS. Además, la encuesta categoriza y discute los principales desafíos técnicos y específicos de la investigación, facilitando así a los docentes la tarea de encontrar temas relevantes en los que puedan centrar sus esfuerzos.



Luego de ejecutada una primera iteración de la estrategia propuesta, se envió una encuesta digital para la validación de la estrategia propuesta y la identificación de buenas prácticas percibidos y elementos a enriquecer. A continuación se presentan los resultados más importantes alcanzados al terminar la implementación de la primera iteración de la estrategia.

Resultados alcanzados en la Iteración 1:

Pregunta 1: ¿Cuál es el grado de facilidad de uso de la PaaS para implementar proyectos ligeros?

De acuerdo a la primera pregunta, en donde se refiere al grado de facilidad de uso de IBM Cloud Lite para la implementación de proyectos ligeros; las opciones que los estudiantes pueden seleccionar son las siguientes: Muy Fácil, Fácil, Poca Facilidad, Difícil y Muy Difícil. Los resultados obtenidos se representan en la Figura 2.

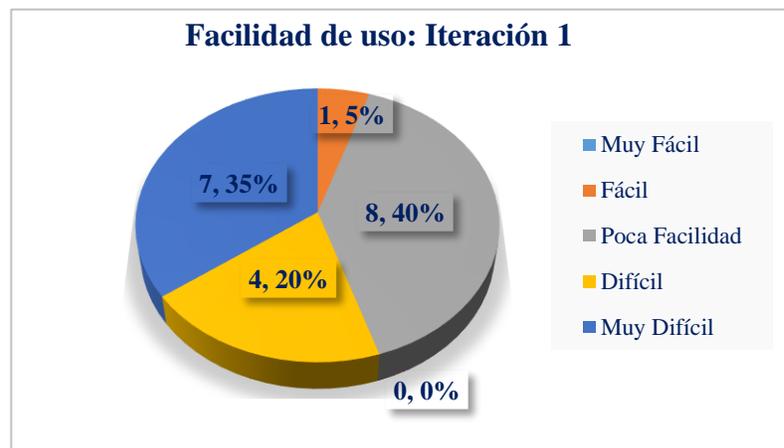


Figura 2. Consideraciones sobre la facilidad de uso de IBM Cloud Lite para la implementación de proyectos ligeros.

Fuente: Elaboración propia.

Respecto a la pregunta 1, el 40% de los estudiantes consideró que la implementación en IBM Cloud tuvo poca facilidad; y el 55% lo consideraron muy difícil. Si se tiene en cuenta que los estudiantes recibieron una capacitación de dos laboratorios, tutoriales y recursos didácticos para el trabajo con la plataforma, esos resultados no se consideran buenos. Es así que en la planificación de la segunda iteración se planificaron 4 laboratorios de entrenamiento con la PaaS. Además, la capacidad de programación de los estudiantes también importa y tiene gran incidencia en este aspecto. Está claro que las habilidades de codificación deficientes tendrán un impacto negativo en el desarrollo de software en general. Es por eso que la mayoría de los estudiantes sufrió la falta de conocimiento de la PaaS y la poca experiencia de programación al mismo tiempo.

En este sentido, y para poder alcanzar los objetivos básicos de la estrategia educativa propuesta, es necesario que los estudiantes se entrenen en el empleo del entorno de desarrollo en la nube, de manera, que durante la implementación



de sus proyectos puedan enajenarse de la tecnología y concentrarse en la capacidad de sus sistemas, en las propiedades arquitectónicas, y en los requerimientos de calidad que este debe cumplir.

Pregunta 2: ¿Después de desarrollado el proyecto, en qué medida considera que la implementación con plataformas PaaS fue útil para la comprensión de conceptos básicos de sistemas distribuidos?

De acuerdo a la segunda pregunta, en donde se refiere a la utilidad de la implementación con plataformas PaaS para la comprensión de conceptos básicos de sistemas distribuidos y diseño arquitectónico, se obtuvieron los siguientes resultados.

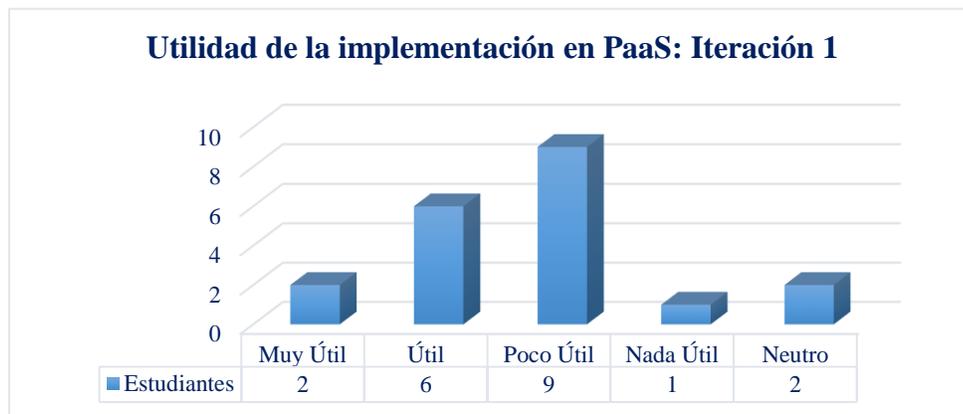


Figura 3. Consideraciones sobre la utilidad de la implementación en PaaS para la comprensión de conceptos.

Fuente: Elaboración propia.

Referente a la utilidad de la implementación en PaaS para la comprensión de conceptos de sistemas distribuidos, el 30% de los estudiantes lo consideraron útil; sin embargo, al 40% le pareció poco útil. El análisis realizado refiere que estos resultados están directamente relacionados a la facilidad de uso percibida de la PaaS, y a las deficiencias en las habilidades de programación. Con estos resultados y las diferentes percepciones emitidas de los estudiantes, se realizó un ajuste en las indicaciones específicas de la implementación de los proyectos para evaluarse en la iteración 2.

Resultados alcanzados en la Iteración 2:

Basado en los resultados obtenidos en la iteración 1, se decidió realizar algunos ajustes para la próxima entrega. En primer lugar, asignar proyectos similares a los estudiantes, de manera que estén familiarizados con los requisitos funcionales y el contexto operacional, para que concentren sus esfuerzos en temas de diseño arquitectónico e implementación.

Sobre la planeación de la capacitación, los profesores se basaron en los resultados de la encuesta y en la evaluación individual de los proyectos implementados, el cumplimiento de los objetivos de aprendizaje, y el nivel de implementación alcanzado. En consecuencia fue necesario incluir aspectos específicos para atender las diferencias



individuales de cada estudiante. Se ajustaron las pautas de la preparación y las instrucciones de los proyectos. Además, los profesores comprobaron y actualizaron los materiales didácticos específicos de PaaS, aun cuando se mantuvieron proyectos similares los mismos proyectos de los estudiantes. Esta actualización debe realizarse continuamente, no solo por la efectividad con que los estudiantes aprenden con los recursos diseñados, sino por los cambios que ocurren en una PaaS, donde los servicios disponibles pueden cerrarse y ser reemplazados por otros sin previo anuncio. En las figuras 4 y 5 se muestran los resultados obtenidos, referente a facilidad de uso y utilidad, una vez realizadas las actualizaciones y ejecutada la segunda iteración:

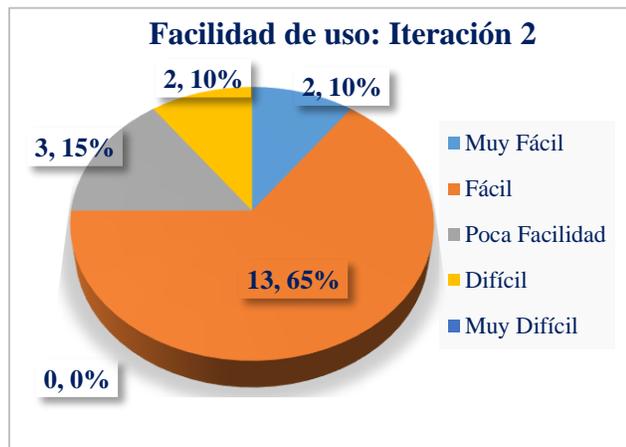


Figura 4. Facilidad de uso: Iteración 2.

Fuente: Elaboración propia.

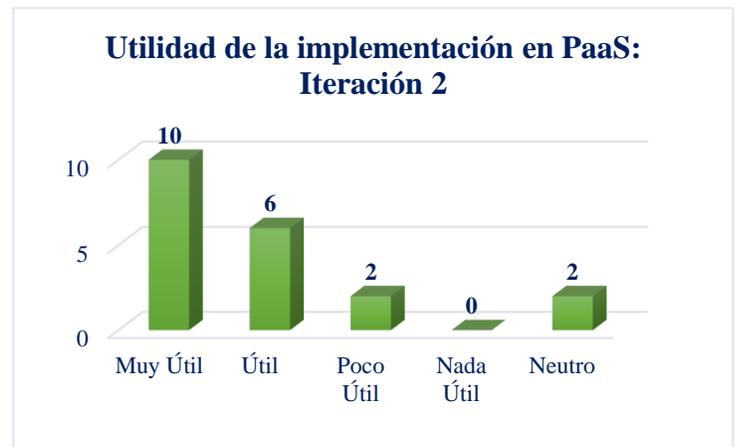


Figura 5. Utilidad de la implementación en PaaS: Iteración 2.

Fuente: Elaboración propia.

Las percepciones de los estudiantes referentes a facilidad de uso y utilidad, aumentaron considerablemente en sentido positivo. Al preguntar si recomendarían a estudiantes de otros periodos el desarrollo de proyectos en la PaaS para complementar los conocimientos teóricos, el 100 % refirió que lo recomendarían. Así mismo, se logró que los estudiantes crearan sus propios equipos de desarrollo para agregar funcionalidades más complejas a sus proyectos y aumentar el tamaño y cantidad funcionalidades. De los proyectos evaluados, tres fueron seleccionados para emplear como recursos de aprendizaje para próximas iteraciones de la estrategia.

Sin embargo, en la evaluación de los proyectos se observaron varias prácticas estudiantiles incompletas. El principal desafío aquí todavía tiende a ser lidiar con problemas de compatibilidad y asimilación tecnológica. Desafortunadamente, tal sobrecarga de colaboración (dominio de sistemas distribuidos e implementación en la nube) parece demasiado alta para algunos estudiantes, de ahí la importancia de actualizar la estrategia a partir de las lecciones aprendidas. De hecho,



incluso los estudiantes que lograron con éxito las tareas planificadas también destacaron este desafío como dificultades en su trabajo. Otro de los desafíos identificados fue el pobre dominio del inglés, que es el idioma en que aparece la mayoría de la documentación relevante. Este desafío requiere inevitablemente extensas discusiones y negociaciones entre los estudiantes.

Discusiones

En el dominio del Aprendizaje Mejorado por Tecnología (TEL) (Kirkwood & Price, 2014), el uso de tecnologías basadas en la nube también se ha identificado como una tendencia clave que permite el acceso a servicios en línea en cualquier lugar y promete escalabilidad, disponibilidad mejorada y ahorro de costos. En la educación, la computación en la nube ofrece propiedades deseables para brindar servicios de aprendizaje electrónico, especialmente en escenarios donde estos servicios requieren un uso intensivo de la computadora (mundos virtuales, simulaciones, transmisión de video).

El uso de PaaS en la educación se discute particularmente en (González-Martínez et al., 2015), y sus principales ventajas se reconocen fundamentalmente en los cursos de programación y arquitectura de software. En esta plataforma, los estudiantes se benefician del entorno de desarrollo listo, de esta manera pueden centrarse en la mejora de sus habilidades de diseño arquitectónico y de programación, sin gastar tiempo y esfuerzos adicionales en la configuración de la pila de tecnología subyacente.

Con el empleo de PaaS, los estudiantes pueden centrarse en el estudio de conceptos abstractos y fomentar el pensamiento de orden superior “aprendiendo a programar, programando”. Sin embargo, es necesario crear recursos adicionales que no enajenen al estudiante de temas técnicos del entorno de desarrollo como la seguridad de la información, la planificación de recursos compartidos, el rendimiento, entre otros.

Los recursos PaaS públicos y gratuitos son satisfactorios para que los estudiantes construyan sistemas de software livianos, mientras que los proyectos de estudiantes pequeños y medianos pueden respaldar el aprendizaje experiencial en el dominio de la arquitectura de software y los sistemas distribuidos. Aunque algunos estudios argumentan que las explicaciones teóricas en clase no pueden revelar la importancia de la arquitectura de software, los resultados de la implementación de la estrategia realzan los beneficios de la capacitación teórica específica como complemento del programa de la asignatura.

Además, la característica pública de los servicios PaaS empleados facilita y fomenta en gran medida el trabajo en equipo de los estudiantes y la colaboración en proyectos adecuados, sin mencionar que las habilidades de trabajo en equipo son particularmente cruciales para que los estudiantes se desarrollen en escenarios de software distribuido.



La estrategia educativa propuesta se ubica en un contexto actual donde aumenta la tendencia de emplear la computación en la nube para aliviar la creciente escasez de presupuesto en educación. Además, en comparación con los recursos en la nube dedicados a la educación, el PaaS público y gratuito no requiere ponerse en contacto con ningún proveedor de la nube ni pasar por ningún papeleo para su uso, lo que incluso minimiza el costo/esfuerzo de administración tradicional a cero. Además, sin tener ninguna restricción de tiempo o lugar de uso, los servicios de PaaS pueden ayudar al aprendizaje permanente de los estudiantes al permitirles acceder y mejorar sus cursos fuera de la universidad y después de graduarse.

Las principales partes interesadas en el proceso de enseñanza aprendizaje, es decir, los estudiantes, los profesionales de la educación, las IES y los profesores de TI, pueden beneficiarse del uso de la computación en la nube, para fortalecer el aprendizaje significativo de los estudiantes, y aumentar su experiencia en implementación y trabajo colaborativo, una vez que salgan egresados a la industria de desarrollo de software.

Conclusiones

Aunque todavía existen límites relacionados con PaaS para ser accedida desde las IES, la estrategia educativa propuesta puede fomentar el aprendizaje experiencial y el aprendizaje colaborativo de los estudiantes. Sobre la base de los objetivos compartidos entre los sistemas distribuidos y el paradigma de la computación en la nube, se recomienda explotar las características y los privilegios de PaaS como entorno de desarrollo de proyectos ligeros de los estudiantes. El análisis de los resultados de la implementación destaca que las funciones de PaaS pueden proporcionar características computacionales avanzadas a los sistemas distribuidos. Asimismo se identificó, que aunque en un entorno de desarrollo real, los sistemas de software suelen ser más grandes y complejos, el desarrollo práctico de proyectos ligeros puede conectar mejor las habilidades de programación de los estudiantes y aumentar el nivel de conocimiento y confianza para implementar sistemas de mayor complejidad.

Los resultados de la investigación muestran que la implementación de proyectos ligeros con plataformas PaaS, tiene gran utilidad para la comprensión de conceptos básicos de sistemas distribuidos y diseño arquitectónico. Para el proceso de enseñanza se recomienda IMB Cloud Lite por ser gratis y contener servicios que son suficientes para este tipo de objetivos.

Conflictos de intereses

Los autores no declaran conflicto de intereses.

Contribución de los autores



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

1. Conceptualización: Bolívar Ramos Mosquera.
2. Curación de datos: Bolívar Ramos Mosquera.
3. Análisis formal: Bolívar Ramos Mosquera.
4. Investigación: Bolívar Ramos Mosquera.
5. Metodología: Bolívar Ramos Mosquera.
6. Software: Bolívar Ramos Mosquera.
7. Validación: Bolívar Ramos Mosquera.
8. Visualización: Bolívar Ramos Mosquera.
9. Redacción – borrador original: Bolívar Ramos Mosquera.
10. Redacción – revisión y edición: Bolívar Ramos Mosquera.

Financiamiento

No se requirió financiamiento externo.

Referencias

- Arulkumar, V., & Lathamaju, R. (2019). Start to finish automation achieve on cloud with build channel: By DevOps method. *Procedia computer science*, 165, 399-405.
<https://www.sciencedirect.com/science/article/pii/S1877050920300405>
- Arunarani, A., Manjula, D., & Sugumaran, V. (2019). Task scheduling techniques in cloud computing: A literature survey. *Future Generation Computer Systems*, 91, 407-415.
<https://www.sciencedirect.com/science/article/pii/S0167739X17321519>
- Cico, O., Jaccheri, L., Nguyen-Duc, A., & Zhang, H. (2021). Exploring the intersection between software industry and Software Engineering education-A systematic mapping of Software Engineering Trends. *Journal of Systems and Software*, 172, 110736. <https://www.sciencedirect.com/science/article/pii/S0164121220301667>
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 319-340. <https://www.jstor.org/stable/249008>
- Ferrer, A. J., Pérez, D. G., & González, R. S. (2016). Multi-cloud platform-as-a-service model, functionalities and approaches. *Procedia computer science*, 97, 63-72.
<https://www.sciencedirect.com/science/article/pii/S187705091632097X>



- Glushan, V., Lavrik, P., & Rybalchenko, M. (2015). Hypergraph model of hierarchical client-server architecture for distributed computing. 2015 9th International Conference on Application of Information and Communication Technologies (AICT),
- González-Martínez, J. A., Bote-Lorenzo, M. L., Gómez-Sánchez, E., & Cano-Parra, R. (2015). Cloud computing and education: A state-of-the-art survey. *Computers & Education*, 80, 132-151. <https://www.sciencedirect.com/science/article/pii/S0360131514001985>
- Kirkwood, A., & Price, L. (2014). Technology-enhanced learning and teaching in higher education: what is ‘enhanced’ and how do we know? A critical literature review. *Learning, media and technology*, 39(1), 6-36. <https://www.tandfonline.com/doi/abs/10.1080/17439884.2013.770404>
- Lee, H.-Y., & Wang, N.-J. (2019). Cloud-based enterprise resource planning with elastic model–view–controller architecture for Internet realization. *Computer Standards & Interfaces*, 64, 11-23. <https://www.sciencedirect.com/science/article/pii/S0920548918301697>
- Li, S., Zhang, H., Jia, Z., Zhong, C., Zhang, C., Shan, Z., Shen, J., & Babar, M. A. (2021). Understanding and addressing quality attributes of microservices architecture: A Systematic literature review. *Information and Software Technology*, 131, 106449. <https://www.sciencedirect.com/science/article/pii/S0950584920301993>
- Li, Z. (2020). Using public and free Platform-as-a-Service (PaaS) based lightweight projects for software architecture education. Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training,
- Ray, P. P. (2016). A survey of IoT cloud platforms. *Future Computing and Informatics Journal*, 1(1-2), 35-46. <https://www.sciencedirect.com/science/article/pii/S2314728816300149>
- Sánchez, P. M. M., & Barrezueta, L. D. R. (2022). Análisis de la información generada para mantener la escalabilidad y persistencia del proceso de desarrollo de software. *Serie Científica de la Universidad de las Ciencias Informáticas*, 15(8), 193-227. <https://publicaciones.uci.cu/index.php/serie/article/view/1137>
- Schümmer, T., Lukosch, S., & Haake, J. M. (2017). Teaching distributed software development with the project method. In *Computer Supported Collaborative Learning 2005* (pp. 577-586). Routledge. <https://www.taylorfrancis.com/chapters/edit/10.4324/9781351226905-76/teaching-distributed-software-development-project-method-till-sch%C3%BCmmer-stephan-lukosch-joerg-haake>
- Schwaber, K., & Beedle, M. (2002). *Agile software development with scrum*. Series in agile software development (Vol. 1). Prentice Hall Upper Saddle River. <https://www.lowrell.com/sites/default/files/webform/cv/agile-software-development-with-scrum-series-in-agile-software-d-ken-schwaber-mike-beedle-2e69aa1.pdf>



- Song, J., Zhang, P., Alkubati, M., Bao, Y., & Yu, G. (2022). Research advances on blockchain-as-a-service: Architectures, applications and challenges. *Digital Communications and Networks*, 8(4), 466-475. <https://www.sciencedirect.com/science/article/pii/S2352864821000092>
- Tsuruoka, Y. (2016). Cloud computing-current status and future directions. *Journal of Information Processing*, 24(2), 183-194. https://www.jstage.jst.go.jp/article/ipsjjip/24/2/24_183/_article/-char/ja/

