

# Tres experiencias docentes con una plataforma en la nube

M. Dolores Ruiz<sup>1</sup> and Francisco Torralbo<sup>2</sup>

<sup>1</sup> Dpto. Ingeniería Informática. Universidad de Cádiz. España

<sup>2</sup> Dpto. Matemáticas. Universidad de Cádiz. España  
{francisco.torralbo, mariadolores.ruiz}@uca.es

**Abstract.** Nowadays, the use of cloud platforms is growing both for research and teach. This paper highlights the advantages and disadvantages of using the CoCalc cloud platform for teaching purposes. More precisely, we describe the platform, its inherent ecosystem and those features that can be of interest for teaching through three different use cases.

**Resumen.** Las plataformas en la nube son cada vez más usadas tanto en investigación como en docencia. En este trabajo se pone de manifiesto las ventajas del uso de una plataforma concreta, CoCalc, para la docencia. En particular, se describe el sistema usado y su ecosistema mediante tres experiencias docentes distintas que ilustran el uso de dicha herramienta.

**Keywords:** Teaching Platform · CoCalc · Jupyter · Sagemath · Python.

## 1 Introducción

Hoy en día es cada vez más habitual el uso de plataformas en la nube tanto para investigación como para uso docente. Esto viene motivado por la ubicuidad del acceso a internet y la versatilidad de las herramientas desarrolladas.

Actualmente se pueden encontrar distintos tipos de plataformas: algunas centradas en la gestión de los recursos docentes (p.e. Moodle [15]) y otras orientadas al desarrollo de software o a la investigación (p.e. Microsoft Azure [5], RCloud [19], Apache Zeppelin [2]... ). Entre estas últimas destacamos la aparición de varias plataformas basadas en la idea de *cuaderno* impulsada por el sistema Jupyter [12] (p.e. CoCalc [6], Anaconda Cloud [1], Cybera [8], Google Colaboratory [11], Google Cloud Datalab [10], Binder [4] ... ).

*Jupyter* es una aplicación web de código abierto que permite crear documentos (llamados *notebooks*, a los que nos referiremos en adelante como cuadernos) que contienen código, ecuaciones, gráficos y texto explicativo. La aplicación se comunica con un *kernel* que evalúa los comandos y devuelve el resultado. Inicialmente se desarrolló para los lenguajes Julia, Python y R de donde procede su nombre. Actualmente es posible usar dicha aplicación web con multitud de lenguajes diferentes (p.e. Sagemath, Octave, Javascript, Haskell, Scala... para una lista completa consultar [14]). La versatilidad de los cuadernos de Jupyter

al permitir en un mismo documento la inclusión de código, texto y figuras lo hace idóneo tanto para la investigación como para la docencia, como puede verse por ejemplo en [3, 9].

Ahora bien, este tipo de plataformas de cómputo en la nube son muy útiles para la docencia interactiva y a distancia pero no pretenden sustituir a las plataformas propiamente docentes, si no más bien complementarlas para facilitar la labor de enseñanza del profesorado, cuando esta se refiere a ejercicios de programación o simplemente el docente quiere ofrecer contenidos interactivos al alumnado.

En este artículo describiremos distintas experiencias docentes usando una de estas plataformas: CoCalc [6] (anteriormente conocido como SageMathCloud). La elección de esta frente al resto ha venido determinada puesto que es la única conocida por los autores que permite usar Sagemath (lenguaje de cálculo simbólico para matemáticas) así como gestionar cursos y alumnos. Es de destacar que esta plataforma ha sido utilizada desde 2013 en más de 200 cursos por miles de estudiantes. Diferentes instituciones de todo el mundo han confiado en esta plataforma como por ejemplo Oxford University, MIT, Sheffield University, Universidad Complutense de Madrid (ver lista completa en [7]) para impartir cursos en áreas muy diversas como son: matemáticas, física, bioinformática, programación, criptografía, computación gráfica, meteorología. . .

## 2 Descripción de la plataforma utilizada

Como hemos puesto de manifiesto en la introducción, CoCalc basa su diseño en Jupyter. Así, los documentos creados en la plataforma serán *cuadernos* que podrán contener texto, ecuaciones mediante  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  y código en distintos lenguajes de programación. Actualmente la plataforma permite la creación de estos cuadernos para Python, R, Sagemath, Octave, Spark, Julia, PostgreSQL, Scala, Singular así como la edición de documentos de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .

Los cuadernos y cualquier documento adjunto se agrupan en *proyectos*. En principio los documentos y cuadernos asociados a un proyecto son de acceso privado (únicamente el propietario del proyecto y los colaboradores que él haya agregado tienen acceso a los mismos). Sin embargo, cualquier cuaderno puede hacerse *público*, es decir, el sistema genera una URL para que cualquier persona pueda acceder (pero no editar) a su contenido.

Los proyectos pueden a su vez convertirse en *cursos*. De esta manera, el sistema permite añadir alumnos al curso para que puedan acceder a los archivos, asignarles tareas con una fecha límite y evaluar sus resultados. Es de destacar que los profesores que gestionan el curso tienen siempre acceso a los cuadernos de cada alumno y pueden editarlos con ellos, contando además con el acceso a un chat para la resolución de dudas en tiempo real (ver figura 1).

De forma opcional, se puede crear un proyecto colaborativo de manera que todos los usuarios (profesores y alumnos) tendrán permiso de edición. Esto puede resultar de utilidad para clases colaborativas en el laboratorio o bien para compartir material con el alumnado para que experimente con él.

## 2.1 Ventajas e inconvenientes de CoCalc

A continuación describiremos las ventajas e inconvenientes de usar este sistema.

En primer lugar podemos destacar las ventajas propias de ser un sistema en la nube: no hay necesidad de instalación por parte del profesor ni del alumnado, es independiente del sistema operativo y todos los usuarios comparten la misma versión del programa y de las librerías. Como desventaja es necesario tener acceso a internet aunque existe la posibilidad de instalar el entorno Jupyter en el ordenador para usar el contenido sin conexión. El inconveniente es la dificultad de esta tarea para un usuario medio así como la necesidad de instalar cada lenguaje de programación a usar de forma individual.

La segunda gran ventaja es la gratuidad del servicio. Existe la posibilidad de pago tanto por parte del profesor como del alumno para para ampliar los servicios como pueden ser: más espacio de almacenamiento, servidor dedicado sin reinicios, cómputos con tiempos de ejecución mayores, acceso a internet a través de los cuadernos, . . . Como contrapartida en su versión gratuita puede no estar disponible en ciertos momentos puntuales y, aunque no existe limitación en el número de alumnos, el rendimiento no es óptimo para su uso en el aula para un número elevado (véase la sección 4),

Otra característica a destacar es el control automático de versiones. El sistema guarda distintas versiones de cada cuaderno y es posible recuperar todo el historial de trabajo de cada alumno para ver la evolución del mismo. Así mismo, la plataforma guarda una copia de todos los archivos de cada proyecto cada cinco minutos mientras se está editando y permite recuperar archivos borrados.

Desde el punto de vista del profesor, CoCalc permite unificar todo el contenido que podría necesitar para su labor docente, combinando en un mismo cuaderno tanto contenido teórico como práctico, ya que un cuaderno puede contener explicaciones en texto y trozos de código ejecutable. Una característica interesante para la labor docente es la posibilidad de exportar los cuadernos en un formato de presentación. Además el profesor puede elaborar material docente tradicional puesto que el sistema dispone de un editor de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . El material elaborado en CoCalc puede ser usado y visualizado en cualquier plataforma basada en Jupyter con el núcleo de SageMath instalado (véase sección 4).

Con respecto a la interacción con los alumnos, la plataforma permite la asignación de diferentes roles para los usuarios. En general uno de los profesores será el propietario del proyecto mientras que el resto de docentes serán colaboradores y todos tendrán permiso de edición sobre todos los archivos del proyecto. Además existe el rol del alumno: este tendrá acceso únicamente al contenido que los profesores estimen oportuno.

Desde el punto de vista de la evaluación, la plataforma permite la gestión de actividades y tareas con fecha límite. Al asignar una tarea a los alumnos los archivos asociados a la misma son copiados en las cuentas personales de cada uno donde tienen acceso de edición. Durante el desarrollo de las tareas, CoCalc permite la revisión y resolución de dudas en tiempo real puesto que el profesor puede acceder al cuaderno que esta editando el alumno en cada momento, corregir errores y darle sugerencias mediante un chat integrado en la plataforma

(ver figura 1). Una vez finalizado el plazo de entrega, el sistema recupera dichos archivos para su evaluación. El profesor puede publicar tanto los resultados de la evaluación como observaciones a la misma (en un documento aparte o dentro del cuaderno del alumno) en la misma plataforma. Una vez finalizada la evaluación, el material generado por los alumnos (ejercicios de evaluación o cualquier tipo de cuaderno de prácticas) puede ser descargado en pdf para su archivado en caso de ser necesario para su posible auditoría. Además, se dispone de una amplia

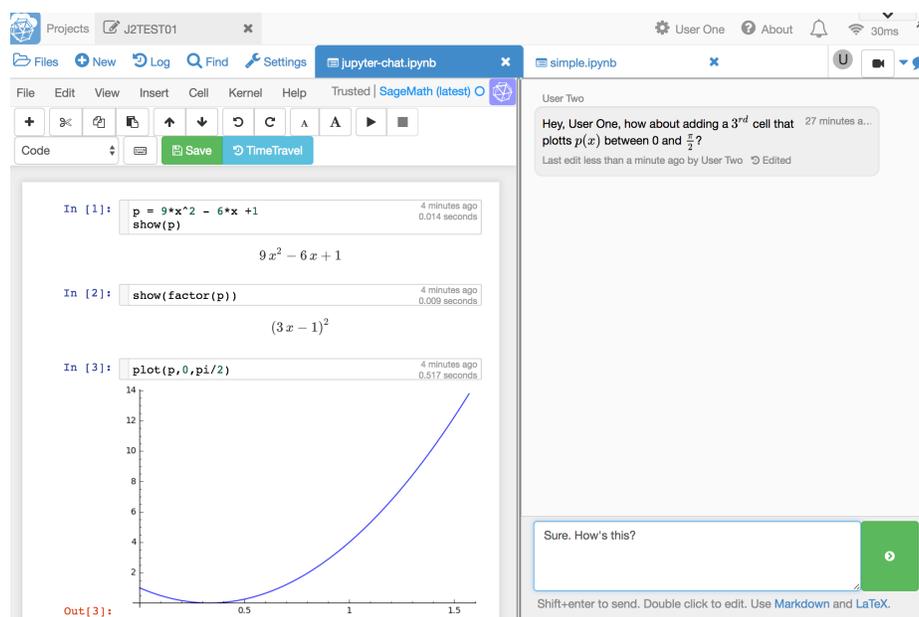


Fig. 1. Interfaz de CoCalc mostrando el chat

información sobre el uso y conexión de los estudiantes. La plataforma indica la fecha de la última conexión de cada estudiante así como la fecha de edición de los cuadernos por parte de este. Esto, junto con el control de versiones de la plataforma, permite monitorizar no sólo el trabajo del alumno sino su evolución y los problemas que ha encontrado.

### 3 Experiencias docentes

En esta sección mostramos tres casos diferentes de uso de la plataforma elegida. Estos ilustran parte del potencial de dicha herramienta para la enseñanza. Se trata de tres casos con objetivos docentes diferentes: en el primero de ellos se pretende únicamente ilustrar conceptos teóricos mediante ejemplos para los que el uso tradicional de la pizarra es insuficiente. En el segundo, la herramienta se

aprovecha para iniciar al alumnado en un sistema de cálculo simbólico. Finalmente, en el tercero se utilizó para impartir una asignatura no presencial.

### 3.1 Experiencia docente 1: Página web interactiva

En muchas situaciones las herramientas disponibles en un aula actual (pizarra y proyector) son insuficientes para enseñar ciertos conceptos pues no permiten al alumno interactuar y, de esa manera, afianzar el conocimiento transmitido en la lección. Esta experiencia se enmarca en un curso de doctorado sobre superficies (área de geometría diferencial).

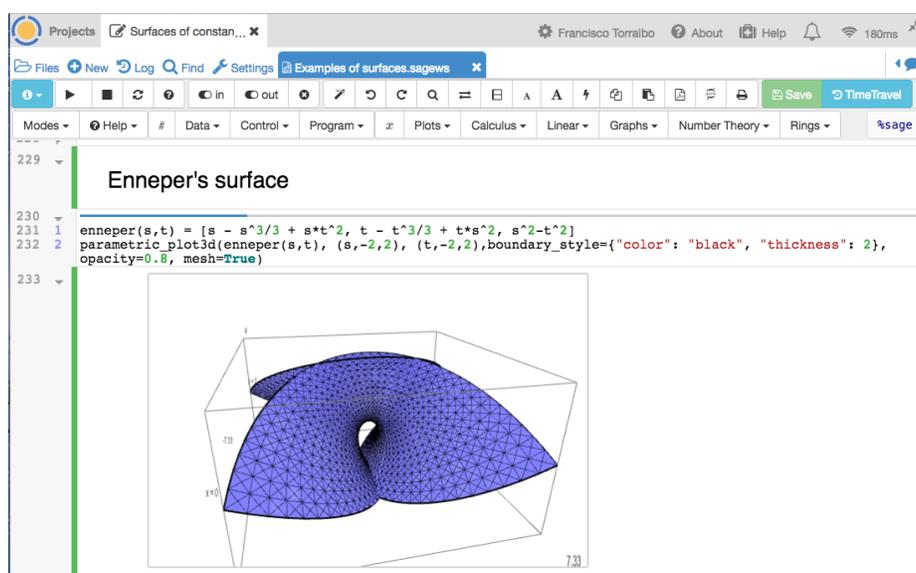


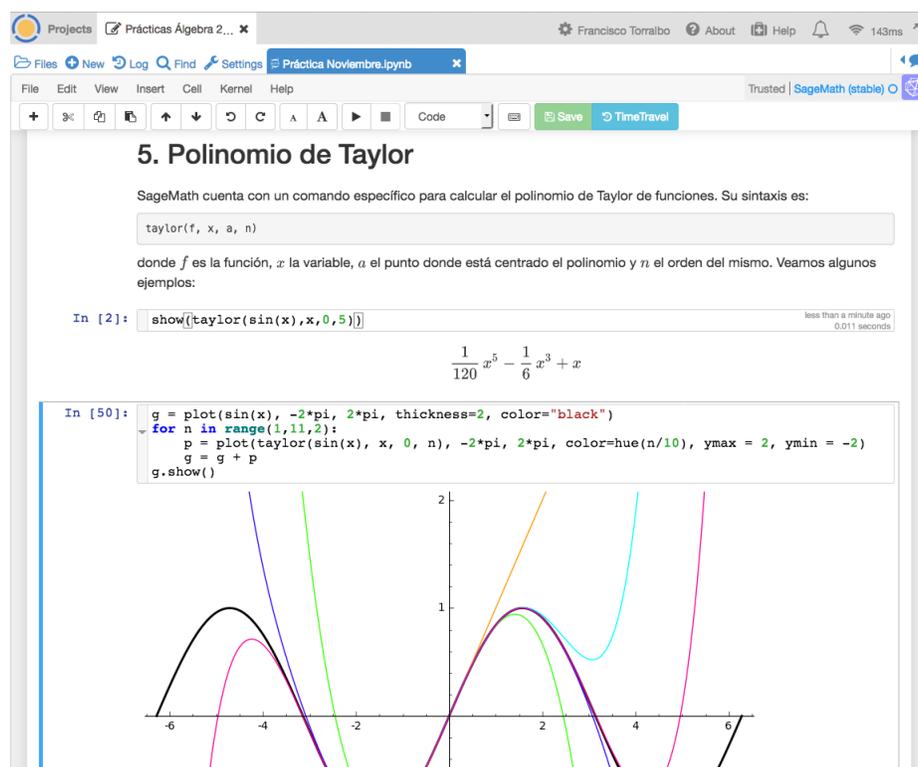
Fig. 2. Experiencia docente 1: Curso de doctorado sobre superficies

En este caso el alumnado no precisaba de una cuenta en la plataforma puesto que el material generado por el profesor estaba disponible de forma abierta. Más precisamente, se creó un cuaderno donde se dibujaron distintos tipos de superficies dependientes de varios parámetros (ver figura 2). Dicho cuaderno fue *publicado* de manera que cualquiera con la URL apropiada y una conexión a internet podía acceder al contenido. En concreto, además de incluir material complementario, el alumno podía manipular las superficies para entender mejor su geometría.

### 3.2 Experiencia docente 2: Prácticas de ordenador individuales

La parte práctica de asignaturas de áreas como las matemáticas o la informática precisa de la instalación y mantenimiento de un software concreto en los labora-

torios. En algunas ocasiones, dicho software no está instalado o bien la versión disponible está obsoleta. Además, dicho software puede requerir al alumnado de una licencia de pago si quiere trabajar fuera del aula de prácticas. En el caso bajo estudio el profesorado se encontraba en esta situación y decidió usar la herramienta elegida en este artículo para la elaboración de unas prácticas de matemáticas en un primer curso de ingeniería.



**Fig. 3.** Experiencia docente 2: Ejemplo del polinomio de Taylor

Durante esta experiencia, se requirió al alumnado registrarse en el sistema. La herramienta se usó tanto para la explicación en el laboratorio (ver figura 3) como para el desarrollo de los problemas propuestos al alumnado y su posterior evaluación. De entre las ventajas descritas en la sección 2.1 destacamos: no hay necesidad de instalación ni configuración por parte de la institución ni por parte del alumno, la gestión de las tareas propuestas con fecha límite y la información sobre el uso de la plataforma por parte de los estudiantes. Esto ayudó a tener un mayor control en la evaluación continua.

### 3.3 Experiencia docente 3: Enseñanza a distancia

Es cada vez más común que parte de la enseñanza de posgrado se realice a distancia dando más flexibilidad a la misma. Para la consecución exitosa de esta ya existen multitud de plataformas distintas enfocadas a la gestión de recursos docentes, siendo la más conocida y extendida Moodle. Sin embargo este tipo de plataformas no son las adecuadas para el desarrollo de unas prácticas con un lenguaje de programación. En esta última experiencia se usó la plataforma CoCalc para ilustrar distintas técnicas de aprendizaje automático (ver figura 4) en un curso de posgrado. En concreto, se usó el lenguaje de programación Python con sus librerías científicas Scipy y Scikit-learn, ambas disponibles en la plataforma. Más precisamente, se elaboraron una serie de prácticas introductorias al lenguaje y a las librerías y a posteriori se propusieron unas actividades prácticas evaluables usando la misma plataforma.

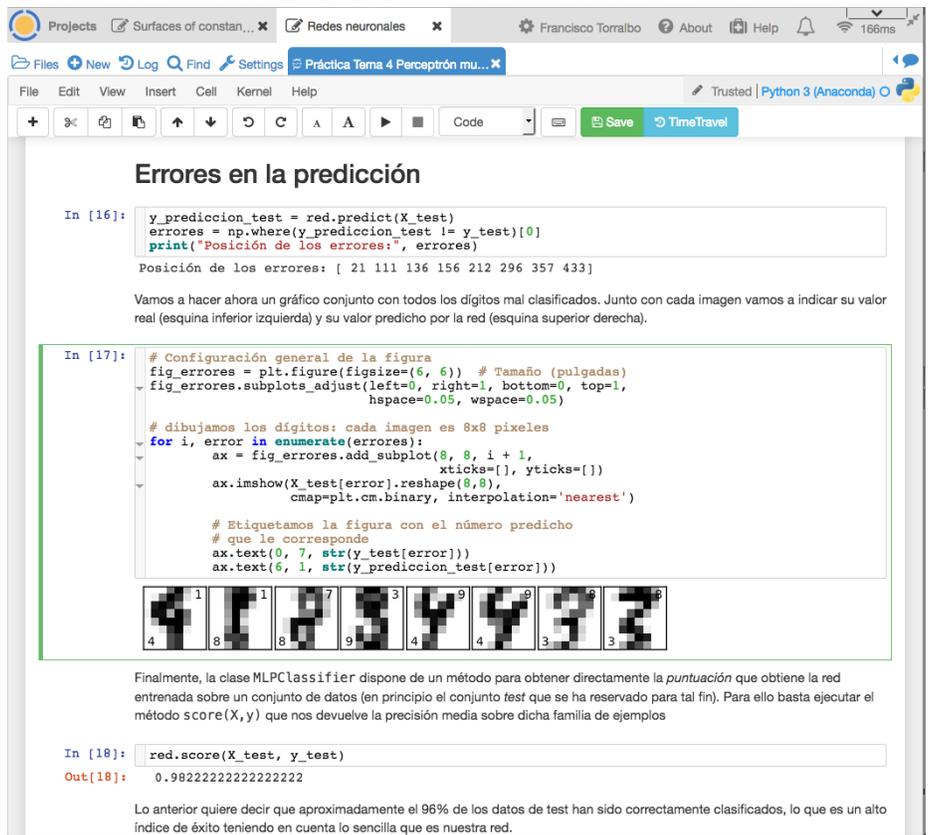


Fig. 4. Experiencia docente 3: Curso de posgrado sobre Redes Neuronales

Entre las ventajas descritas anteriormente destacamos: todos los usuarios (profesores y alumnos) compartían la misma versión del lenguaje y de las librerías, evitando problemas derivados del uso de versiones diferentes; y la resolución de dudas en tiempo real mediante el chat que ofrece la plataforma. Además, debido al software usado en esta experiencia (Python y sus librerías científicas) existía la posibilidad para el alumno de instalar dicho software de forma gratuita (p.e. mediante la distribución Anaconda [1]) y trabajar sin conexión a internet.

## 4 Limitaciones y alternativas

Durante las distintas experiencias docentes se han observado varias limitaciones de la plataforma usada.

La primera limitación que queda patente es la imposibilidad de elaborar cuestionarios autoevaluables. Esta herramienta, disponible en múltiples plataformas docentes como Moodle, no se encuentra actualmente disponible en CoCalc. Esta herramienta sería de mucha utilidad para evaluar contenidos teóricos que suelen presentarse en las primeras fases de la enseñanza de una asignatura. Esto, a su vez, permitiría al profesor conocer el nivel de conocimiento de la materia del alumnado antes de comenzar las prácticas. Pero como ya pusimos de manifiesto en la introducción, este tipo de plataformas de cálculo no pretenden sustituir a las ya clásicas de gestión de recursos docentes que sí poseen esta funcionalidad. Así, un uso adecuado de cada una de ellas permitirá que se complementen. No obstante, hay prevista una extensión en CoCalc [17], que permitirá al profesor evaluar de forma automática ejercicios de programación mediante el diseño de tests automáticos.

La segunda limitación solo existe en su versión gratuita y es la imposibilidad de hacer cálculos intensivos o instalar librerías adicionales en caso de ser necesarias para una tarea concreta. En un uso reciente de la plataforma los autores han detectado un descenso considerable en el rendimiento de la plataforma que hace desaconsejable su uso con una cuenta gratuita. No obstante, en las prácticas elaboradas no se ha precisado de un alto proceso de cálculo y la mayoría de software y librerías usadas hoy en día están disponibles en sus versiones más recientes.

Una alternativa que solucionaría estas dos limitaciones es la adquisición de un servicio para cursos en CoCalc, que supondría un gasto semestral fijo que depende del número de alumnos. Otra opción es la instalación y mantenimiento de un servidor de Jupyter mediante el software JupyterHub [13] en el que habría que instalar el sistema de cálculo simbólico SageMath como un núcleo de Jupyter. Esta segunda opción permitiría realizar cálculos intensivos (de acuerdo a las prestaciones del servidor) y además existe la oportunidad de instalar la extensión de Jupyter Nbgrader [16] que permite implementar una forma de autoevaluación para problemas de programación usando Python. Como tercera alternativa, si la institución tiene un convenio con Google, es usar el servicio Google Colaboratory [11] que permite ejecutar y compartir cuadernos de Jupyter con una interfaz parecida a Google Drive.

La tercera limitación encontrada es la evaluación de prácticas grupales, ya que no existe una herramienta que permita crear o evaluar una actividad para un grupo de usuarios que permita además poder evaluar la participación de cada uno de ellos.

Una alternativa interesante a tener en cuenta es el servicio Binder [4]. Este permite generar un entorno jupyter a partir de un repositorio en GitHub sin la necesidad, por parte del alumno, de instalar software ni tener ninguna cuenta de usuario en un servicio en la nube. Como contrapartida, aunque el contenido generado es interactivo para el alumnado, no hay posibilidad de guardar los cambios introducidos en la sesión. Esta última alternativa podría enmarcarse dentro de la primera experiencia docente descrita en la sección 3.

## 5 Conclusiones

Las tres experiencias docentes descritas muestran el potencial tanto de esta plataforma en la nube (así como de otras similares que se han ido indicando) para su uso en diferentes escenarios docentes.

En todos los casos descritos se usó la versión gratuita de la misma durante los cursos académicos 2015–2016 y 2016–2017 y en ningún momento esto supuso algún problema en la ejecución de las actividades o afectó a la calidad docente de los cursos impartidos. No obstante, la plataforma CoCalc ha cambiado su política de servicio y actualmente la experiencia de uso de una cuenta gratuita ha empeorado. Esto se debe a la necesidad de hacer rentable el servicio en la nube debido a los gastos de mantenimiento que esto genera.

La mejor alternativa encontrada hoy día por los autores es o bien adquirir un servicio para cursos en CoCalc, que supondría un gasto semestral fijo que depende del número de alumnos, la instalación y mantenimiento de JupyterHub, que supondría un desembolso inicial por parte de la institución en hardware y mano de obra especializada en su puesta en marcha e instalación, o bien usar el servicio Google Colaboratory [11], que obligaría tanto al alumnado como al profesorado disponer de una cuenta de google si no existe convenio previo de la institución con dicha compañía (véase la sección 4).

Los profesores, gracias al control de versiones y a la monitorización ofrecida por el sistema, pudieron detectar cuando alguno de los ejercicios propuestos habían sido compartidos entre los alumnos, lo que supone una ventaja añadida al uso de esta plataforma.

Finalmente, aunque aquí se han descrito las bondades de CoCalc para fines docentes a través de la experiencia de los autores, queremos poner de manifiesto que también es adecuada para su uso en investigación como plataforma colaborativa.

## Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad a través de los proyectos de investigación TIN2015-64776-C3-1-R (primera autora) y MTM2014-52368-P (segundo autor).

## References

1. *Anaconda: Open data science platform powered by Python*. <https://anaconda.org/>. Disponible online. Último acceso 26/07/2017.
2. Apache software foundation. *Apache Zeppelin: A web-based notebook that enables interactive data analytics.* <https://zeppelin.apache.org/>. Disponible online. Último acceso 27/07/2017.
3. F. Aznar et al. Ipython Notebook: Herramienta para integración de teoría y práctica en Ingeniería y Arquitectura. *XIV Jornadas de Redes de Investigación en Docencia Universitaria: Investigación, innovación y enseñanza universitaria: enfoques pluridisciplinares* ISBN 978-84-608-7976-3, págs. 1678-1691, (2016).
4. Binder. *Binder: Turn a GitHub repo into a collection of interactive notebooks*. <https://mybinder.org/>. Disponible online. Último acceso 30/05/2018.
5. Microsoft Corporation, *Azure: servicios integrados en la nube*. <https://azure.microsoft.com/>. Disponible online. Último acceso 26/06/2017.
6. *CoCalc: Collaborative Calculations in the Cloud*. <https://cocalc.com/>. Disponible online. Último acceso 16/06/2017.
7. *CoCalc in teaching*. <https://github.com/sagemathinc/cocalc/wiki/Teaching>. Disponible online. Último acceso 18/06/2017.
8. *Cybera: Jupyter 'All-in-One' Science Platform*. <https://www.cybera.ca/>. Disponible online. Último acceso 27/06/2017.
9. E. Díaz et al. Nuevas herramientas de software libre para la corrección automática de ejercicios complejos. [Proyecto de Innovación Docente] (2016) <http://eprints.ucm.es/35311/>. Disponible online. Último acceso 05/06/2018.
10. *Google Cloud Datalab*. <https://cloud.google.com/datalab/>. Disponible online. Último acceso 27/06/2017.
11. *Google Colaboratory*. <https://colab.research.google.com/>. Disponible online. Último acceso 05/06/2018.
12. *Project Jupyter*. <http://jupyter.org/>. Disponible online. Último acceso 27/06/2017.
13. *JupyterHub*. <https://github.com/jupyterhub>. Disponible online. Último acceso 05/06/2018.
14. *Jupyter kernels*. <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>. Disponible online. Último acceso 26/06/2017.
15. *Plataforma de aprendizaje Moodle*. <https://moodle.org/>. Disponible online. Último acceso 26/06/2017.
16. *Nbgrader: A system for assigning and grading notebooks*. <https://github.com/jupyter/nbgrader>. Disponible online. Último acceso 11/08/2017.
17. *Cocalc Jupyter extension similar and compatible with nbgrader*. <https://github.com/sagemathinc/cocalc/issues/1929>. Disponible online. Último acceso 30/05/2018.
18. *Python programming language*. <https://www.python.org/>. Disponible online. Último acceso 26/06/2017.
19. *RCloud: The Social Coding Environment for Big Data Analytics & Visualization*. <http://rcloud.social/>. Disponible online. Último acceso 27/07/2017.