

# Uso de Jupyter notebooks y nbgrader para ofrecer retroalimentación en una asignatura de programación

Juan Gutiérrez-Aguado   Ignacio García-Fernández   Manolo Pérez-Aixendri  
Departamento de Informática  
Universitat de València  
Avda de la Universitat, s/n 41600 Burjassot (Valencia)

juan.gutierrez@uv.es

ignacio.garcia@uv.es

manolo.perez@uv.es

## Resumen

Presentamos una experiencia del uso de Jupyter Notebooks en una asignatura de Programación usando Java, de tercer curso de Ingeniería Telemática, con el objetivo de reducir el plazo de la realimentación proporcionada tras las tareas. El alumnado dispone de dos formas de realizar los notebooks: desde un navegador durante las clases, accediendo a un despliegue de JupyterHub sobre Kubernetes; o mediante una máquina virtual. Se han diseñado 11 notebooks sobre aspectos percibidos tradicionalmente como complicados. Para evaluar la utilidad de los notebooks en el proceso de enseñanza-aprendizaje, se ha diseñado una encuesta con respuestas cuantitativas y cualitativas para las que se ha realizado un análisis temático. Los resultados muestran que el alumnado considera útil el uso de los notebooks y valora positivamente la incorporación de retroalimentación en los mismos.

## Abstract

We present an experience of the use of Jupyter Notebooks in a programming subject that uses Java. This subject is taught in the third year of the Ingeniería Telemática degree. The goal is to reduce the time required to give feedback after performing programming tasks. Students can perform the notebooks by using a browser during the classes through JupyterHub deployed on Kubernetes; or whenever they want by using a virtual machine. Eleven notebooks have been designed on topics that have been perceived as complicated over the past years. To evaluate the usefulness of the notebooks in the learning process, a survey has been passed and analyzed using thematic analysis. Results show that students consider the usage of the notebooks useful, and have considered positively the addition of feedback.

## Palabras clave

Participación activa, retroalimentación, Jupyter notebooks, nbgrader

## 1. Introducción

La realización de ejercicios prácticos es un apoyo imprescindible en las sesiones de teoría para conseguir una mejor comprensión de los conceptos. Aunque, idealmente, los ejercicios los harán los estudiantes, la experiencia nos indica que la mayoría preferirá atender a la explicación o utilizar las notas de clase (cuando no una mera fotografía de la pizarra) para repasar los ejercicios fuera del aula. Los ejercicios propuestos como deberes fuerzan a una mayor participación –fuera del aula– pero plantean al profesorado el reto de dar realimentación en un plazo breve para que ésta sea efectiva. En el caso de las asignaturas de programación, la evaluación de los ejercicios puede requerir plazos largos que demoran esta realimentación. Sin embargo, existen múltiples herramientas que permiten realizar una primera comprobación de los ejercicios y generar una realimentación rápida de forma automática, tanto específicas para la docencia como provenientes de las metodologías de desarrollo de software que pueden ser aplicadas al ámbito docente [1, 5, 6].

Este trabajo presenta una propuesta para la utilización de la herramienta Jupyter Notebooks, junto con el complemento nbgrader, con la finalidad de realizar trabajos tanto en el aula como fuera de ella y ofrecer retroalimentación inmediata a los estudiantes en asignaturas de programación. Nuestra hipótesis es que una dinámica de resolución de ejercicios más participativa y con realimentación instantánea mejorará la motivación y la implicación del alumnado en los ejercicios prácticos en el aula y, además, ayudará a los estudiantes a regular de forma autónoma su aprendizaje.

Los objetivos de este trabajo forman parte de un conjunto más amplio de retos, planteados como un Pro-

yecto de Innovación Docente financiado por la Universitat de València (UV-SFPIE\_PID-2079770) que continúa el trabajo de cursos anteriores (proyecto UV-SFPIE\_PID-1641550), en los que se desplegó un repositorio de control de versiones para su uso en diferentes asignaturas [2]. Con el proyecto actual se intenta dar un paso más, añadiendo herramientas que permitan dar retroalimentación inmediata sobre el desempeño en asignaturas de programación.

En lo que resta de la sección 1 se describe la asignatura en la que se ha llevado a cabo la experiencia. En la sección 2 se presentan las herramientas seleccionadas para conseguir los objetivos planteados. La sección 3 incluye los detalles sobre la implementación de las actividades docentes. En concreto se indica cómo se han diseñado las tareas y cómo puede el alumnado realizarlas (tanto en el aula como fuera de ella). La metodología seguida para la realización de las tareas se presenta en la sección 4. Por último, la evaluación de los resultados obtenidos tras llevar a cabo la propuesta se discute en la sección 5.

### 1.1. Asignatura involucrada

La asignatura *Programación* de 6 ECTS (4 teórico/prácticos y 2 de laboratorio) se imparte en el primer cuatrimestre del tercer curso del Grado de Ingeniería Telemática de la Universitat de València. El alumnado matriculado en esta asignatura ha cursado dos asignaturas previas relacionadas con la programación usando el lenguaje C++: Informática de primer curso (algoritmos, programación estructurada, programación procedural, tipos estructurados y ficheros) y Ampliación de Informática en el segundo curso (introducción a la orientación a objetos y la herencia, tipos abstractos de datos y algoritmos de ordenación y búsqueda). La asignatura Programación usa el lenguaje Java y se organiza en 5 temas donde los contenidos van desde una revisión de conceptos de orientación a objetos, la programación concurrente y aplicaciones en red.

A lo largo de los cursos en que se ha impartido esta asignatura se han elaborado 21 boletines (en total 49 páginas disponibles en formato PDF) con ejercicios cortos que se resuelven en las clases teórico/prácticas. El objetivo de estos boletines, que cubren la totalidad de aspectos tratados en clase, es poner en práctica los conceptos mostrados durante la exposición de la teoría y fomentar la reflexión y la discusión sobre ellos. De los 5 temas en los que se divide el curso, el primero cuenta con 10 boletines y en el resto se realizan entre 1 y 4.

La metodología usada en las clases teórico/prácticas ha consistido (hasta el curso 21/22) en la exposición de la teoría por parte del profesorado y en la resolución de un boletín de problemas sobre el aspecto explicado. Durante la realización del boletín se intenta que el

alumnado participe y exponga sus dificultades. Sin embargo, a lo largo de los años se ha observado que una parte del alumnado no se siente motivado por este tipo de actividad y en lugar de intentar resolver el problema planteado, se limita a realizar una foto una vez ha sido resuelto perdiendo una oportunidad de aprendizaje.

En consecuencia, al inicio del curso 22/23 se optó por explorar otras vías de resolución de ejercicios, tanto en clase como fuera ella, para ver su efecto en la motivación y en la implicación del alumnado. Además, se planteó como objetivo que la herramienta utilizada incorporara retroalimentación sobre el desempeño, de tal manera que permitiera al alumnado regular de forma autónoma el proceso de aprendizaje. A continuación se describe detalladamente la estrategia implantada.

## 2. Uso de Jupyter con nbgrader para proporcionar retroalimentación del desempeño

Como base para implantar el sistema de realización de ejercicios se ha escogido el sistema de *notebooks* Jupyter<sup>1</sup>. Este sistema permite lanzar una interfaz web con un editor en línea que interpreta código. El editor permite elaborar documentos (*notebooks*) como una secuencia de celdas (unidades de información) que son, básicamente, de dos tipos:

- *Markdown*: celdas que contienen texto
- *Code*: celdas que contiene código que se puede ejecutar

Las celdas que contienen texto permiten introducir formato utilizando el lenguaje de marcado *Markdown*. En el caso de las celdas que contienen código, cuando se ejecutan muestran a continuación de la propia celda el resultado de interpretar o ejecutar el código (incluidos los errores, si se producen). Este sistema permite elaborar documentos interactivos sobre programación y proponer ejercicios. Aunque originalmente Jupyter surge como un entorno interactivo para ejecutar Python, actualmente existen núcleos de ejecución para multitud de lenguajes, incluido Java. Jupyter permite, por tanto, la elaboración de tareas para los estudiantes, redactando el enunciado en celdas de texto y dejando código fuente y notas sobre las tareas a realizar en celdas de código, que deben ser completadas como parte de la tarea.

Sobre esta tecnología, se propone la creación de tareas con *nbgrader* [3] que permite generar *notebooks* con dos tipos adicionales de celdas:

- *Autograded answer*: celdas en las que el alumnado debe escribir código

<sup>1</sup><https://jupyter.org/>

- *Autograder test*: celdas que ejecutan pruebas sobre el código escrito y que tras ejecutarse indican si se pasan o no las pruebas.

La figura 1 (superior) muestra un ejemplo de diseño de una tarea por parte del profesorado donde se puede observar que hay una celda del tipo *Autograded answer*. Al exportar el *notebook*, el código que aparece entre los comentarios `//BSOL` y `//ESOL` se sustituye por un comentario que indica al alumnado que debe insertar su código. Se puede observar también que hay una celda del tipo *Autograder test* donde se comprueba si la solución proporcionada por el alumnado pasa las pruebas. Esto permite dar retroalimentación al alumnado. Las celdas del tipo *Autograder test* no son editables una vez se genera el *notebook* para su distribución. La figura 1 (inferior) muestra la versión exportada que se distribuiría al alumnado.

### 3. Detalles de la implementación

#### 3.1. Diseño de las tareas

Las tareas son desarrolladas por el profesorado desde su máquina, ejecutando un contenedor que tiene habilitado Jupyter con un núcleo de Java y la extensión nbgrader. Una vez diseñados los *notebooks* de la tarea, estos se exportan para generar la versión que se distribuirá al alumnado y que no contiene las soluciones<sup>2</sup>.

Para la distribución de los *notebooks* se ha optado por usar un repositorio Git, donde el profesorado sube la versión de los *notebooks* generada por nbgrader para el alumnado<sup>3</sup>. El alumnado puede clonarlos para obtener una copia local, o puede actualizarlos si ya ha clonado previamente repositorio (las instrucciones para realizar estas dos acciones se proporcionan al alumnado con antelación). El flujo de trabajo a seguir, por parte del profesorado, para desarrollar una nueva tarea es por tanto la siguiente:

1. Lanzar el contenedor con Jupyter, el núcleo de Java y nbgrader y acceder desde un navegador.
2. Abrir *Formgrader*, que ofrece una interfaz para gestionar las tareas, y crear una nueva.
3. Redactar los *notebooks* de la tarea usando los tipos de celdas que ofrece nbgrader.
4. Una vez creados todos los *notebooks* de la tarea, se exporta generando la versión para el alumnado.
5. Subir la versión para el alumnado de tarea al repositorio Git.

<sup>2</sup>En el repositorio <https://github.com/jgutierrez2r/jupyter-java-nbgrader> está disponible el código para generar tanto la imagen del contenedor del profesorado como la del alumnado junto con un *notebook* de ejemplo

<sup>3</sup>En esta versión se ha sustituido la solución por comentarios donde se indica que se escriba el código

Para realizar las tareas, el alumnado necesita disponer de la infraestructura necesaria para ejecutar los *notebooks*. En el caso de una configuración desde cero, esto requiere instalar y configurar software en sus máquinas, por lo que se han ofrecido dos soluciones alternativas: una que permite la ejecución de los *notebooks* sin necesidad de instalar nada (se accederá mediante el navegador a una infraestructura configurada) y otra que consiste en descargar, e instalar en su propia máquina, una máquina virtual ligera que tiene todo lo necesario. A continuación se describe cómo se han preparado ambas instalaciones<sup>4</sup>

#### 3.2. Realización de las tareas usando computación en la nube

Se ha realizado un prototipo de despliegue de JupyterHub sobre una infraestructura privada de computación en la nube basada en OpenStack<sup>5</sup> para permitir la realización de las tareas desde cualquier dispositivo durante las clases. Para realizar este despliegue se han utilizado las siguientes herramientas (con una considerable adaptación a nuestra infraestructura de la guía de Loïc Gouarin<sup>6</sup>:

- *Terraform*<sup>7</sup>: permite crear y configurar un clúster de máquinas virtuales a partir de ficheros de configuración (infraestructura a partir de código). El clúster consta de 5 máquinas worker (con 18 GB RAM y 5 vCPUs) y un nodo máster (con 10 GB RAM y 4 vCPUs). La figura 2 muestra el clúster de máquinas virtuales creadas.
- *Ansible*<sup>8</sup>: para la instalación de Kubernetes<sup>9</sup> sobre el clúster de máquinas virtual, la instalación del controlador de OpenStack sobre Kubernetes (que permite crear volúmenes para el almacenamiento de bloques de forma dinámica).
- *Helm*<sup>10</sup>: para la instalación de JupyterHub.

Es importante indicar que fue necesario realizar varias iteraciones para ajustar parámetros del despliegue (como por ejemplo los recursos asignados al clúster virtual o el fichero de configuración de JupyterHub).

La figura 3 muestra una captura del acceso a los *notebooks* usando este despliegue de JupyterHub desde un navegador (sin necesidad de ejecutar nada en la má-

<sup>4</sup>También sería posible ejecutarlos en Google Colab, pero en este caso habría que incluir una celda inicial en todos los *notebooks* en la que se instale Java y el núcleo que permite enviar el código a Java. En nuestro caso no hemos optado por esta solución ya que no hay convenio entre Google y la UV (aunque el alumnado a título individual podría usarlo).

<sup>5</sup><http://openstack.org>

<sup>6</sup><https://links.uv.es/V2VF4hh>

<sup>7</sup><https://www.terraform.io/>

<sup>8</sup><https://www.ansible.com/>

<sup>9</sup><https://kubernetes.io/>

<sup>10</sup><https://helm.sh/>

**Predicate**

La interfaz funcional `Predicate<T>` define el método `boolean test(T t)`.

Escribe una expresión lambda válida para crear un `Predicate<Estudiante>` que compruebe si un estudiante es menor de 20 años:

In [3]:

```
import java.util.function.Predicate;

Predicate<Estudiante> pe =
    // BSOL
    e -> e.getEdad() < 20;
    // ESOL
```

Una vez tengas el código, ejecuta la siguiente celda para comprobar si tu implementación es correcta:

In [4]:

```
%maven org.assertj:assertj-core:3.20.0
import static org.assertj.core.api.Assertions.*;

Estudiante e = new Estudiante("Antonio", 25);

// Invocamos al método test implementado en la expresión lambda
assertThat(pe.test(e)).isEqualTo(false);
```

**Predicate**

La interfaz funcional `Predicate<T>` define el método `boolean test(T t)`.

Escribe una expresión lambda válida para crear un `Predicate<Estudiante>` que compruebe si un estudiante es menor de 20 años:

In []:

```
import java.util.function.Predicate;

Predicate<Estudiante> pe =
    // YOUR CODE HERE
```

Una vez tengas el código, ejecuta la siguiente celda para comprobar si tu implementación es correcta:

In []:

```
%maven org.assertj:assertj-core:3.20.0
import static org.assertj.core.api.Assertions.*;

Estudiante e = new Estudiante("Antonio", 25);

// Invocamos al método test implementado en la expresión lambda
assertThat(pe.test(e)).isEqualTo(false);
```

Figura 1: Imagen superior: Ejemplo de diseño de un notebook de una tarea donde se puede ver el uso de celdas *Autograded answer* y de *Autograded test*. Imagen inferior: Previsualización de la tarea tal y como se muestra al alumnado. Como se puede observar, el código de la solución no se muestra y en su lugar aparece un comentario indicando que se debe añadir el código.

quina desde la que se realiza el acceso). Como se puede observar, se está accediendo a una dirección IP pública (que es la dirección IP flotante asignada al nodo máster tal y como se muestra en la figura 2). Para acceder se solicita un usuario (obtenido de la cuenta de correo de la UV y matriculado en la asignatura).

### 3.3. Realización de las tareas usando una máquina virtual local

Para favorecer la realización de los *notebooks* cuando el alumnado está fuera del aula, se ha creado y distribuido una máquina virtual ligera (que usa como base Alpine Linux), totalmente configurada para que al ejecutarse redirija las peticiones del puerto 8888 de la máquina anfitrión a la máquina virtual donde se ejecuta el servicio de Jupyter. Esta imagen de máquina virtual ocupa 2.2 GB y se ha proporcionado al alumnado en

Nombre de la instancia	Nombre de la imagen	Dirección IP	Sabor	Par de claves	Estado	Zona de Disponibilidad	Tarea	Estado de energía
<input type="checkbox"/> jhub-k8s-node-3	ubuntu-xenial	10.0.0.14	jhub-worker	jhub-keys	Activo	nova	Ninguno	Ejecutando
<input type="checkbox"/> jhub-k8s-node-1	ubuntu-xenial	10.0.0.12	jhub-worker	jhub-keys	Activo	nova	Ninguno	Ejecutando
<input type="checkbox"/> jhub-k8s-node-5	ubuntu-xenial	10.0.0.13	jhub-worker	jhub-keys	Activo	nova	Ninguno	Ejecutando
<input type="checkbox"/> jhub-k8s-node-4	ubuntu-xenial	10.0.0.5	jhub-worker	jhub-keys	Activo	nova	Ninguno	Ejecutando
<input type="checkbox"/> jhub-k8s-node-2	ubuntu-xenial	10.0.0.11	jhub-worker	jhub-keys	Activo	nova	Ninguno	Ejecutando
<input type="checkbox"/> jhub-k8s-master-1	ubuntu-xenial	10.0.0.10 IPs flotantes: 147.156.86.8	jhub-master	jhub-keys	Activo	nova	Ninguno	Ejecutando

Figura 2: Clúster desplegado sobre la infraestructura de computación en la nube

formado OVA (1.1 GB). De este modo, los estudiantes solo deben ejecutar la máquina virtual y acceder desde su navegador a `http://localhost:8888` teniendo así una experiencia similar a la plataforma en línea proporcionada durante las sesiones de clase.

#### 4. Metodología para la realización de las tareas

Para la realización de un *notebook* en el aula, se pide al alumnado que acceda a la URL donde se ejecuta el nodo máster. Para cada conexión, JupyterHub lanza un servidor (desplegado como un *pod* en el argot de Kubernetes) para ejecutar *notebooks* y le asigna su volumen (almacenamiento de bloques) persistente. Si es el primer acceso, entonces el volumen debe crearse antes de ser asignado. Al contenedor que se ejecuta en el *pod* se le inyecta una variable de entorno a fin de facilitar la obtención de los *notebooks* del repositorio Git.

El alumnado puede realizar el *notebook* desde cualquier dispositivo electrónico (tableta, portátil o incluso con el móvil aunque no es lo más recomendable ya que hay que introducir código). En la infraestructura actual no se ha implantado ningún sistema de autenticación, que compruebe que las credenciales proporcionadas con válidas. Por ello, y para liberar los recursos para otras asignaturas, al finalizar la tarea se deshabilita el acceso, de forma que no está disponible fuera del horario de clase.

Además, si el alumnado desea realizar algún *note-*

*book* fuera del aula, puede usar la máquina virtual proporcionada, descrita en la sección 3.3, sin ninguna restricción. Gracias a la redirección de puertos configurada por defecto, una vez se lanza la máquina virtual, el entorno que encuentran los estudiantes es similar al que se les proporciona en clase.

#### 5. Toma de datos y evaluación de la propuesta

En esta sección se presenta el diseño metodológico usado para la valoración de la propuesta. En concreto, se presenta la encuesta realizada y cómo se han analizado los resultados obtenidos.

Se ha diseñado un conjunto de preguntas (4 cuantitativas, 2 con respuesta a seleccionar entre varias opciones y 3 cualitativas) y la toma de resultados se ha realizado durante una sesión de clase de teoría al final del cuatrimestre. Las preguntas cuantitativas (en escala de Likert de 5 puntos donde 1 indica una valoración baja y 5 una valoración alta) son:

1. Indica si consideras útil el poder realizar los notebooks en línea (desde el navegador), sin tener en cuenta los problemas técnicos de este curso.
2. Indica si consideras útil el disponer de una máquina virtual para la ejecución de los notebooks fuera del aula.
3. Indica el grado en el que consideras que los notebooks te han ayudado en el aprendizaje.
4. Indica si consideras útil que los notebooks integren pruebas para comprobar si las respuestas son

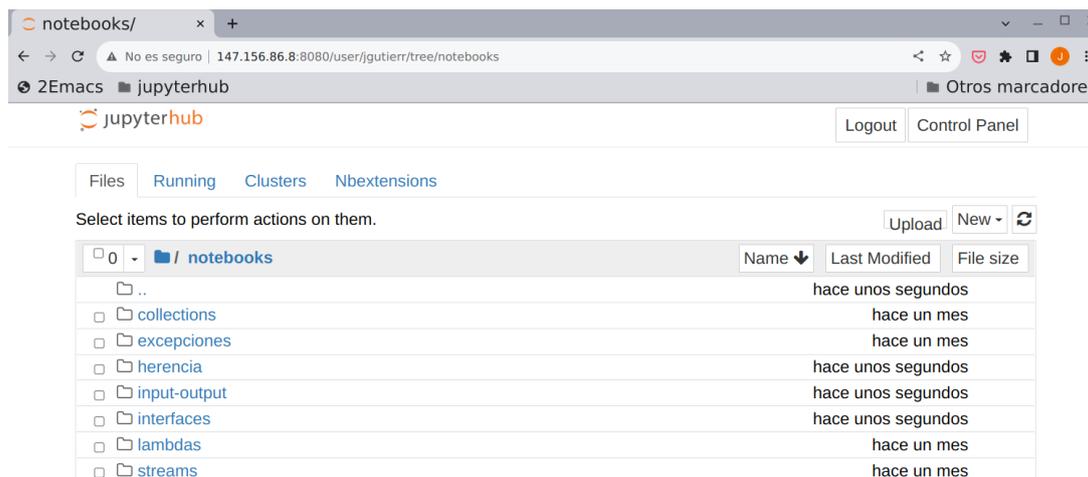


Figura 3: Acceso a los *notebooks*, usando el despliegue de JupyterHub, desde el navegador (sin ejecutar nada en la máquina desde la que se accede).

o no correctas.

Hay dos preguntas para seleccionar entre los valores propuestos:

- Indica los notebooks que has realizado por tu cuenta (fuera del aula). Las opciones posibles de respuesta a esta pregunta son: Ninguno, Algunos, Muchos, Todos
- ¿Mantendrías el uso de los notebooks en la asignatura? Las posibles opciones son Sí o No.

Finalmente, se proponen tres preguntas con respuesta abierta:

- Indica sobre qué partes de la asignatura sería conveniente desarrollar más notebooks.
- Indica aquellos aspectos que te hayan parecido positivos respecto al uso de los notebooks en la asignatura.
- Indica aquellos aspectos que te hayan parecido negativos sobre el uso de los notebooks en la asignatura.

Se ha diseñado un cuestionario para la toma de datos y se ha alojado en la plataforma usada en la UV (basada en Moodle), recogándose un total de 25 cuestionarios. Para el análisis de datos se descartan 5 cuestionarios de estudiantes que indican que no han utilizado la herramienta (la asistencia a clase no es obligatoria; por tanto entendemos que no han realizado ningún cuestionario ni en el horario de clase ni fuera de ella). De los 20 restantes, 13 han contestado que han realizado algunos *notebooks*, 4 indican que han realizado muchos y 3 contestan que los han realizado todos.

## 5.1. Análisis de las respuestas cuantitativas y cerradas

En cuanto a la pregunta de si consideran útil poder realizar los *notebooks* en línea (desde el navegador), se han obtenido los resultados que se muestran en la figura 4-a). Como se puede observar, un 80 % da una puntuación de 4 o más. La figura 4-b) muestra los resultados obtenidos para la pregunta de si consideran útil el disponer de una máquina virtual para la ejecución de los *notebooks* fuera del aula. Un 68.4 % de las respuestas da una puntuación superior o igual a 4 y solo un 10.5 % considera que no es útil. Por tanto, en futuras ediciones del curso se mantendrán las dos opciones.

En cuanto a la utilidad de los *notebooks* para el aprendizaje, se han obtenido los resultados que se muestran en la figura 4-c). Se observa que un 68.4 % de las respuestas otorgan una valoración de 4 o más y en este caso la valoración más baja en esta pregunta ha sido un 3. Esto muestra que, claramente, el alumnado ha considerado útil el uso de los *notebooks*.

Finalmente, en cuanto a si se considera útil que los *notebooks* integren pruebas para comprobar si las respuestas son correctas, los resultados obtenidos se muestran en la figura 4-d). En este caso un 84.2 % ha valorado con un 4 o más la incorporación de pruebas que dan retroalimentación. Es decir, el alumnado percibe como positiva la retroalimentación.

Respecto a la pregunta 6 sobre si mantendrían el uso de los *notebooks* en la asignatura, se han obtenido 17 respuestas afirmativas (aproximadamente un 90 %) y 2 respuestas negativas (aproximadamente un 10 %).

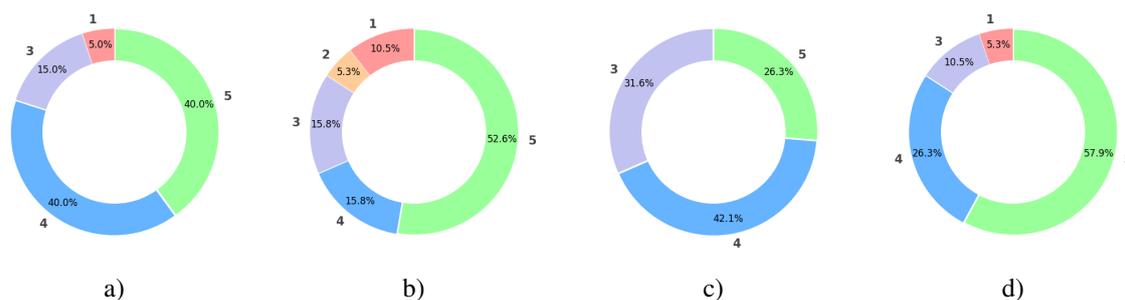


Figura 4: Resultados de las preguntas cuantitativas. a) Pregunta 1; b) Pregunta 2; c) Pregunta 3; y d) Pregunta 4.

## 5.2. Análisis de las respuestas cualitativas

La pregunta 7 pretende identificar las partes de la asignatura que requieren el desarrollo de más *notebooks*. El alumnado ha manifestado que sería interesante disponer de más *notebooks* en prácticamente todos los temas de la asignatura. Por tanto, de cara al siguiente curso se desarrollará un catálogo más amplio. Para las preguntas 8 y 9, se ha realizado un análisis temático de las respuestas [4]. Los temas obtenidos han sido consensuados entre los tres autores del presente trabajo. En la pregunta 8, sobre los aspectos que les han parecido positivos respecto al uso de los *notebooks* en la asignatura, se han encontrado los siguientes temas:

- *Incorporación de la realimentación en los notebooks*: se han encontrado 9 respuestas que se pueden clasificar en este tema. Por ejemplo: «Me ha ayudado bastante a reforzar mis conocimientos de clase, ya que se podía llevar de forma práctica y con feedback. Esto ayuda bastante a la hora de enseñarse por uno mismo». Otra respuesta es «El notebook te da feedback sobre si es correcto o no y en este último caso da pistas sobre donde está el problema.»
- *Permite poner en práctica aspectos teóricos*: en este tema se pueden clasificar 7 respuestas. Por ejemplo: «Ayudan a practicar de cara al examen». Otro ejemplo: «Es un ejemplo práctico sobre los aspectos que hemos visto en clase y esto me ha ayudado a entenderlos mejor». Otra respuesta: «Permite la práctica adicional de los contenidos estudiados en clase.»
- *Permite un trabajo guiado y específico*: en este tema se clasifican 3 respuestas. Por ejemplo: «Esta muy bien que no tengo que implementar todo el código, con implementar la parte clave, ya funciona. Así puedo practicar lo que me interesa, sin perder el tiempo en otras cosas».
- *Facilidad de uso*: en este tema se pueden clasificar 2 respuestas, como por ejemplo: «Fácil uso e intuitivo».
- *Herramienta motivadora y que mejora el aprendi-*

*zaje*: en este tema se pueden clasificar 3 respuestas, como por ejemplo: «Pienso que deberían utilizarse más a menudo, ya que hace que te implique y muestre interés por la asignatura.».

En la pregunta 9, sobre aquellos aspectos percibidos como negativos sobre el uso de los *notebooks* en la asignatura, se han obtenido los siguientes temas:

- *Problemas con el acceso a los notebooks*: en este tema se han clasificado 7 respuestas entre las que destacan: «Al principio, aclararte con la máquina virtual para abrir los notebooks», «Difícil de instalarlo, y de actualizarlos», «El acceso es complicado». Se observa que algunos estudiantes han tenido algunos problemas para acceder a los *notebooks*. Esto nos indica que hay que mejorar la documentación en los siguientes cursos.
- *Falta potenciar su uso*: una respuesta indica «Que no se han apoyado, para usarlos, es decir, sería mucho mejor si se potenciara más su uso». En este caso no queda claro si se refiere a potenciar más el uso en la asignatura o que se debería extender su uso al resto de asignaturas relacionadas con la programación.
- *La retroalimentación que ofrece no es muy informativa*: dos respuestas se han clasificado en esta categoría. Un ejemplo de respuesta es «Estaría bien mejorar la calidad de las respuestas del sistema ( si es posible ) para que no mostrara org@asset y que mostrara alguna salida por pantalla del programa o algún mensaje personalizado». Es cierto que la salida que proporciona el test no es muy informativa (es el `toString()` de la aserción). En futuras ediciones se buscará cómo mejorar esta retroalimentación.
- *No está la solución*: dos respuestas se han agrupado en este tema. Una respuesta indica: «Quizá se debería poder ver las respuestas en caso de que no se pueda llegar a un resultado correcto». No se ha dado la solución ya que se han resuelto y comentado en clase.
- *Requiere una solución incremental*: los *notebooks* a veces tienen preguntas que dependen de solu-

ciones anteriores, esto ha supuesto un problema según se indica en una respuesta: «Si me atasco en una pregunta en algunos casos no puedo continuar con los siguientes apartados».

- *No se ha explicado cómo rehacer los notebooks*: una respuesta indica que «si haces los notebooks luego no los puedes volver a hacer porque se ha guardado lo que hemos hecho». Obviamente, hay varias formas de conseguir esto pero no las conocen y por tanto se debe documentar.

## Agradecimientos

Este trabajo forma parte de un Proyecto de Innovación Docente financiado por la Universitat de València (UV-SFPIE\_PID-2079770).

## 6. Conclusiones y trabajo futuro

En este trabajo se ha presentado una experiencia de uso de *notebooks* que incorporan retroalimentación sobre el desempeño en una asignatura de programación que usa el lenguaje Java. Se han descrito las herramientas utilizadas para diseñar los *notebooks*, el despliegue realizado usando JupyterHub para facilitar la realización de los *notebooks* y decisiones adoptadas para crear un flujo de trabajo lo más simple posible (tanto para el profesorado como para el alumnado). La propuesta se ha valorado mediante una encuesta pasada al alumnado en clase con 4 preguntas cuantitativas, 2 preguntas con respuesta cerrada y 3 preguntas con respuesta abierta. Finalmente, se han obtenido 20 respuestas que se han analizado (estadísticamente las cuantitativas y mediante un análisis temático las cualitativas). Los resultados muestran que el alumnado ha valorado positivamente no solamente el uso de los *notebooks* sino también que éstos incorporen retroalimentación y que mayoritariamente consideran que el uso de los *notebooks* se debe mantener en cursos posteriores. El análisis cualitativo ha puesto de manifiesto algunos aspectos que se deben mejorar de cara a los futuros cursos: problemas en el acceso, mejora de la realimentación y de la documentación proporcionada.

En cuanto al trabajo futuro, se explorarán tecnologías alternativas para la ejecución de notebooks como Zeppelin. No obstante, en un análisis inicial se ha observado que el tamaño de la imagen del contenedor de

Zeppelin es 5 veces mayor que el de las propuestas en este trabajo (7.6 GB frente a 1.4 GB). Así mismo, se propone añadir nuevos *notebooks* en los temas indicados por el alumnado, se estudiará la integración de los *notebooks* en la evaluación de la asignatura, se analizará la implantación de esta herramienta en otras asignaturas relacionadas con la Programación y se implementará autenticación de usuarios. Se plantea también como trabajo futuro

## Referencias

- [1] Jesús Aransay y Jónathan Heras. Fin de un viaje infinito: integración continua para prácticas de programación. En *XXVI Jornadas de Enseñanza Universitaria de la Informática (JENUI)*, 2020.
- [2] J. Gimeno, I. García-Fernández, M. Pérez, I. Coma, F. Martínez-Gil, J. García-Calderaro, S. Casas, J.V. Albert, E. Durá, M. Lozano, I. García-Pereira, C. Portalés, y F. Barber. Generalización del uso de sistemas de control de versiones para los trabajos prácticos del grado de multimedia. En *VII Jornades d'innovació educativa de la Universitat de València*, 2021.
- [3] Project Jupyter, Douglas Blank, David Bourgin, Alexander Brown, Matthias Bussonnier, Jonathan Frederic, Brian Granger, Thomas Griffiths, Jessica Hamrick, Kyle Kelley, M Pacer, Logan Page, Fernando Pérez, Benjamin Ragan-Kelley, Jordan Suchow, y Carol Willing. Nbgrader: a tool for creating and grading assignments in the jupyter notebook. *Journal of Open Source Education*, 2(11):32, 2019.
- [4] Moira Maguire y Brid Delahunt. Doing a thematic analysis: A practical, step-by-step guide for learning and teaching scholars. *All Ireland Journal of Higher Education*, 9(3), 2017.
- [5] Samiha Marwan, Ge Gao, Susan Fisk, Thomas W. Price, y Tiffany Barnes. Adaptive immediate feedback can improve novice programming engagement and intention to persist in computer science. En *ICER '20*, p. 194–203. Association for Computing Machinery, 2020.
- [6] Boban Vesin, Katerina Mangaroska, Kamil Akhuseyinoglu, y Michail Giannakos. Adaptive assessment and content recommendation in online programming courses: On the use of elo-rating. *ACM Trans. Comput. Educ.*, 22(3), 2022.