

Evaluación de una metodología novel basada en cuadernos interactivos para el aprendizaje de la programación en Python

Alejandro Benito, Rodrigo Santamaría, Roberto Therón
GRupo de Investigación en InterAcción y eLearning
Departamento de Informática y Automática
Universidad de Salamanca
abenito, rodri, theron@usal.es

Resumen

La enseñanza de la programación a alumnos de nivel preuniversitario sin experiencia en pensamiento computacional supone un reto que necesita de métodos didácticos novedosos especialmente concebidos para un público que en los últimos años ha modificado radicalmente la forma en la que recibe información. Esta contribución describe la experiencia de adaptar y evaluar, en el contexto de una clase de alumnos de primer curso del Grado en Estadística (N=43), una metodología y un ecosistema de herramientas interactivas orientadas al aprendizaje de la programación y la ciencia de datos. Bajo este enfoque, la metodología experimental concebida implementa técnicas como la programación por pares y TDD (Test-Driven Development), lo cual, como se pudo comprobar a través de las evaluaciones realizadas, tuvo un efecto positivo en el aprovechamiento del tiempo dedicado a la asignatura y en la satisfacción general percibida por el alumnado sobre la impartición de los contenidos de la guía docente. En línea con otras investigaciones actuales, los materiales didácticos fueron puestos a disposición de la comunidad docente de manera abierta y gratuita, lo cual se espera contribuya a mejorar la metodología en futuras experiencias.

Abstract

Teaching programming to undergrad students without previous experience in computational thinking poses a challenge that calls for novel didactic methods oriented explicitly towards this kind of audience. In our contribution, we describe the experience of adapting and evaluating a novel methodology and its associated ecosystem of tools in a classroom of 43 first-year students (N=43) of statistics. Following recommendations found in the literature, the experimental methodology implements pair-programming and TDD (Test-

Driven Development) techniques intending to improve both the teaching and learning experiences. From the evaluation, we could learn that this had a positive impact on the students' performance and their general satisfaction with the course. Finally, the educational materials of this research were collected and put online under a Creative Commons license so that other scientific peers can replicate our work in other future didactic experiences.

Palabras clave

Enseñanza de la programación, recursos para docencia y evaluación, entornos interactivos de aprendizaje, JupyterHub

1. Introducción

Cada vez más, el manejo de herramientas informáticas supone un factor clave en el desempeño de cualquier tarea profesional cualificada. Mientras que en el pasado esta concepción se limitaba principalmente, salvo en algunos pocos casos, al denominado campo de la ofimática y al manejo de alguna herramienta software específica de la disciplina, en el presente la tendencia está cambiando: El aumento en volumen y en complejidad de los conjuntos de datos requiere de profesionales excelentes en el manejo de ordenadores a niveles de maestría impensables hasta hace sólo unos pocos años. Esto es aún más patente en las disciplinas STEM, tradicionalmente más ligadas a la informática.

En este proyecto se pone en práctica una arquitectura de software basada en cuadernos interactivos de Python junto a una metodología de aprendizaje adaptada de experiencias previas. De manera general, un *notebook* o cuaderno interactivo es un documento electrónico que puede ser leído como si se tratase de una revista, pero también ejecutado como un programa de

ordenador. A pesar de que esta idea no es nueva, el enfoque de *notebook* computacional ha cobrado fuerza recientemente en el mundo de la enseñanza de las ciencias de la computación [6, 9, 12]. Existe por tanto un interés dentro de la comunidad docente en alejarse de la configuración tradicional que promueve la comunicación unidireccional profesor-alumno, que en muchas ocasiones dificulta o imposibilita completamente el aprendizaje por parte del estudiante. Recientes avances en la computación en la nube y en la virtualización facilitan enormemente las tareas de configuración necesarias para poner en marcha los llamados MOOC (Massive Open Online Courses), los cuales acarrear importantes ventajas a la hora de facilitar la tarea del aprendizaje al alumno [2]. Nuestra metodología incorpora elementos de evaluación tomados de otras experiencias docentes pasadas [2, 4, 10, 11].

La experiencia docente de la que esta investigación resulta se llevó a cabo entre los meses de septiembre de 2017 y enero de 2018 con 43 alumnos del curso “Informática I” del Grado en Estadística de la Universidad de Salamanca (USAL). Mediante la adaptación de una metodología novel y un ecosistema de herramientas interactivas para el aprendizaje creados recientemente en la Universidad de Berkeley en California [6] se impartieron los contenidos prácticos estipulados en la guía académica, que se resumen como “Que el alumno llegue a comprender la problemática asociada a la implementación de programas, tanto desde el punto de vista teórico como práctico, así como las estructuras de datos y control básicas en cualquier lenguaje de programación”. Para ello, se dedicó un tiempo de clase de 2 horas semanales durante el primer cuatrimestre del curso en los laboratorios de informática de la Facultad de Ciencias de la Universidad de Salamanca.

El resto de este artículo está organizado como sigue: En la Sección 2 se explican los objetivos a conseguir mediante la adaptación del ecosistema educativo mencionado previamente. En la sección 3 se comentan los resultados de una encuesta realizada al comienzo del curso cuyo objetivo era evaluar el nivel de conocimientos en STEM, alfabetización digital y otros parámetros de interés en el alumnado. En la Sección 4 se describe la arquitectura software, *JupyterHub*, que permitió la aplicación de los métodos propuestos por la metodología experimental que se describe en la Sección 5. En la sección 6 se describen los resultados de la evaluación final que se realizó al final del curso, en la que se midieron parámetros como la satisfacción del alumnado con la metodología experimental o la manera en la que los alumnos usaron las tecnologías que se pusieron a su disposición. Finalmente, esta en la Sección 7 extraemos algunas conclusiones importantes de nuestro trabajo y apuntamos algunas líneas de trabajo futuras de interés para el lector.

2. Objetivos

Los objetivos primarios que se marcaban al comienzo del proyecto eran los siguientes: (a) Acelerar el proceso de aprendizaje de la programación por medio del empleo de *Jupyter Notebooks* durante las sesiones prácticas de la asignatura, (b) seguimiento completo del progreso y satisfacción del alumnado y (c) evaluación y diseminación satisfactorias de los resultados obtenidos a lo largo del curso. Además de la consecución de estos objetivos, se estableció como prioritario el intentar mejorar mediante automatización el tiempo invertido por alumnos y docentes en tareas como la el envío, recogida y evaluación de ejercicios o la confección y comunicación de calificaciones, tareas que tradicionalmente se venían realizando a través del correo electrónico o la plataforma Moodle. Esta característica deseable se consiguió mediante la inclusión del módulo “nbgrader”[6] o la creación de un *script* capaz de enviar calificaciones en formato HTML a los alumnos de manera automática (ver Sección 5).

En cuanto al seguimiento del alumnado durante el curso, observamos que, empleando una automatización del proceso de revisión de prácticas, sería posible proveer a los participantes con realimentación rápida y adecuada sobre su progreso en el curso. Esto, junto a las sesiones de tutorías, permite un seguimiento adecuado del alumnado y una toma rápida de decisiones durante el curso que se pensó redundarían en la calidad general del curso. Finalmente, se planeó hacer hincapié en la generación de contenidos públicos y abiertos que puedan ser consultados en Internet. Para ello, todo el código fuente relacionado con el curso está disponible en un repositorio de código con fin de ser consultado y/o reutilizado en el futuro, bien por los mismos docentes, o bien por otros distintos. Aparte de los objetivos primarios comentados, se marcaron una serie de objetivos secundarios relativos que también consideramos en la evaluación propuesta de la metodología experimental: (1) Promoción del trabajo en equipo y la colaboración entre los alumnos, (2) participación proactiva de los alumnos en las horas de clase, (3) trabajo fuera de las horas de clases a través de los recursos en línea accesibles a través de la arquitectura propuesta y (4) incentivación del auto-aprendizaje y estimulación de la curiosidad por medio de cuadernos interactivos.

3. Descripción del alumnado

La primera herramienta de importancia que se empleó en el curso fue una encuesta anónima que motivó un análisis previo de las expectativas y otros parámetros de interés acerca del alumnado. Con el objetivo de conocer las motivaciones del alumnado y la experiencia previa en el uso de ordenadores y programación,

to presenta las siguientes ventajas:

1. El servicio autentica a los distintos usuarios usando sus credenciales de la USAL a través del servicio single sign-on.
2. Cada alumno lanza una instancia personalizada de *Jupyter* que se emplea de manera transparente. Esto asegura que todos los alumnos ejecutan el mismo entorno, solucionando muchos de los problemas anteriormente mencionados.
3. Cada instancia es individualmente asignada a cada estudiante y está aislada de las del resto de alumnos. Las acciones que se realicen en cada instancia no afectan por tanto al funcionamiento de las demás.
4. Las distintas instancias cuentan además con su propio espacio personal que el alumno puede usar para guardar el progreso de su trabajo, anotar código y revisar posteriormente. Este hecho facilita la tarea de estudio y seguimiento continuado de los contenidos explicados en clase por parte del alumno.
5. Es un sistema multiplataforma que funciona en navegadores web. Por lo tanto, sólo es necesario un ordenador con acceso a internet y un navegador web moderno para acceder al mismo.
6. El procesamiento ocurre en el lado del servidor, por lo que la máquina del alumno no se cuelga debido a bucles infinitos lanzados por error.
7. El profesorado puede lanzar actualizaciones durante el curso que los alumnos reciben de manera instantánea en sus respectivas instancias.
8. Esta metodología hace posible distribuir contenido de distinta naturaleza progresivamente a los alumnos durante el curso (ver Sección 5).
9. De manera análoga al punto anterior, resulta sencillo recoger el contenido enviado por los alumnos en tiempo real fuera o dentro de las sesiones prácticas en el aula.
10. La arquitectura es escalable a un número mucho mayor de alumnos, por lo que se tiene la seguridad de que no supondrá una limitación a la hora de evolucionar la metodología en otros contextos de aplicación.

Los alumnos accedieron a dicho servidor durante el curso, que fue desplegado en una máquina configurada a tal efecto dentro del dominio de la universidad, en concreto en el Instituto de Biología Funcional y Genómica de la USAL. La máquina contaba con 64 GB de memoria RAM, procesador de 2.4 GHz y ejecutaba el sistema operativo Linux Ubuntu 16.04 LTS. El servicio JupyterHub se desplegó siguiendo la guía oficial del software y otros recursos [1]. El servidor estuvo en funcionamiento hasta Marzo de 2018, momento en que se salvó una copia de seguridad del mismo y se elimi-

naron sus contenidos.

5. Metodología propuesta

La metodología didáctica propuesta se organizó en torno a tres pilares fundamentales: *Evaluación continua*, *prácticas obligatorias* y *examen práctico de programación* en ordenador. Se describen en el resto de esta sección las particularidades de cada una de ellas: Las sesiones prácticas semanales se dividen en dos partes y se llevan a cabo en equipos compartidos por parejas (de acuerdo a buenas prácticas de programación en parejas). En un estudio de referencia del año 2002, [8] demuestran las virtudes de este enfoque en un curso de nivel introductorio a la programación como era nuestro caso y por este motivo se eligió esta forma de trabajo. El lenguaje de programación elegido fue Python en su versión 3.

La *evaluación continua* se llevó a cabo durante las sesiones de prácticas del curso, con una duración de 2 horas en horario de tarde. Durante la primera parte de la sesión se realizaba una introducción a los contenidos a través de un cuaderno interactivo creado a tal efecto. A este cuaderno se le denominó “lab”, para diferenciarlo de los trabajos prácticos (“tareas”) que se realizarían en la segunda parte de la sesión. Al inicio de cada una de las sesiones, se publicaba un enlace en la plataforma Moodle para que los alumnos pudieran “clonar” el fichero fuente *ipynb* del lab en su instancia personal de *Jupyter Notebook*. Dicha instancia se encontraba ejecutándose en el servidor de la asignatura y de manera transparente, los contenidos eran capturados desde un repositorio de código alojado en una URL de la plataforma colaborativa *GitHub* y copiados a la carpeta personal del alumno.

Para afianzar la comprensión de los contenidos se propusieron una serie de *tareas* relacionadas con el contenido explicado en la primera mitad de la sesión, que los alumnos tendrían que trabajar durante el resto de la clase de manera colaborativa (tanto con su pareja como con el resto de alumnos). Las dudas que surgían se resolvían de manera pública en la clase. Las tareas se ponían a disposición de los alumnos a través del módulo de *JupyterHub* “nbgrader” [6]. Este módulo facilita la creación y corrección de tareas en un cuaderno Jupyter. Permite a los docentes crear de manera sencilla tareas basadas en cuadernos que incluyen ejercicios de código y también de respuesta en modo texto. nbgrader también proporciona una interfaz al profesor que permite corregir rápidamente los trabajos enviados por los alumnos. Las tareas se corrigen manualmente a través de dicha interfaz y, al terminar el proceso, se genera un fichero HTML con los resultados de la calificación para cada alumno (Figura 2). Ya que el módulo no contaba con la opción de auto-

Ejercicio 2:

Escribe una función python que actúe como el método update() explicado en clase, pero que en vez de sobrescribir valores para claves repetidas los combine en una lista.

```
In [4]: Student's answer (Top)
def une(diccionario_1, diccionario_2):
    diccionario_3 = {} #Definimos el diccionario vacío
    for curso in diccionario_1:
        if curso in diccionario_2: #Buscamos las claves que estén en ambos diccionarios
            diccionario_3[curso] = [diccionario_1[curso], diccionario_2[curso]] #Añadimos al nuevo diccionario la lista con los valores de las claves que estén en ambos diccionarios
        else: diccionario_3[curso] = diccionario_1[curso] #Añadimos las claves únicas del primer diccionario
    for curso in diccionario_2:
        if curso not in diccionario_1: #Buscamos las claves del segundo diccionario que no estén en el primero
            diccionario_3[curso] = diccionario_2[curso] #Añadimos las claves únicas del segundo diccionario
    return diccionario_3 #La función devuelve el nuevo diccionario

Comments:
¿Podrías usar un sólo diccionario? Esta sería la opción más correcta.
```

Figura 2: Comentarios en formato HTML enviados a un alumno por correo electrónico. Un script creado a tal efecto por los responsables de la asignatura se encargaba de recoger los comentarios y enviar los ficheros a los alumnos por correo electrónico.

matizar el envío de los ficheros a los alumnos, se creó un pequeño *script* a tal efecto. La dinámica empleada para la entrega de tareas fue el siguiente: Los alumnos que no terminaban la tarea en la hora dedicada a tal efecto, debían completarla en horario fuera de clase y entregarla a través de la plataforma con anterioridad al comienzo de la siguiente sesión. El repositorio de código fue actualizado paulatinamente durante el curso con nuevos **labs**. La última versión del mismo está disponible de forma pública bajo la licencia Creative Commons en la URL: <https://github.com/grialusal/materiales-informatica-i>.

Se propusieron además *dos prácticas obligatorias*, de dificultad y duración más elevadas que las tareas descritas previamente. El objetivo de estas prácticas era que los alumnos conectasen los diferentes bloques didácticos en programas más complejos. En concreto se propusieron dos ejercicios, cuyo código fuente se subió al repositorio de código una vez acabado el curso. Las prácticas se integraron también en el módulo *nbgrader* como en el caso de las tareas semanales pero con la diferencia de que en este caso se siguieron técnicas de TDD para motivar al alumnado a conseguir subobjetivos con el fin de mejorar el aprendizaje [3]. Bajo este enfoque los alumnos progresan a través de una serie de tests incrustados en el cuaderno que comprueban que la implementación cumple el enunciado de la práctica y, por lo tanto, pueden conocer su nota *aproximada* final incluso antes de recibir realimentación por parte del profesor. En la parte derecha de la Figura 3 pueden consultarse los *tests* que se diseñaron para la primera práctica ². El *examen práctico* de la asignatura se realizó en los ordenadores de un aula de informática de la Facultad de Ciencias y tuvo una hora

²<https://github.com/grialusal/materiales-informatica-i/blob/master/practicas/practica-1/practica-1.ipynb>

```
1 #Prueba
2 ##En números impares
3 assert(encuentra_maximo('62\n28') == -1)
4 ##En 67
5 assert(encuentra_maximo('62\n27') == 67)
6
7 #5x5
8 ##Columna arriba-abajo
9 assert(encuentra_maximo('62959\n29678\n42887\n22742\n22222') == 96873)
10 ##Columna abajo-arriba
11 assert(encuentra_maximo('62456\n29678\n42886\n22742\n22222') == 24875)
12 ##Fila izquierda-derecha
13 assert(encuentra_maximo('62656\n29679\n42886\n22742\n22222') == 29679)
14 ##Fila derecha-izquierda
15 assert(encuentra_maximo('62656\n79679\n42886\n22742\n22222') == 97697)
16
17 #10x10
18
19 assert(encuentra_maximo('6743605871\n182095321\n8958054851\n7486519455\n1389386144\n4672017818\n9082874626\n46728') == 99881)
20
21 ##Diagonales
22
23 assert(encuentra_maximo('92656\n79679\n42886\n22782\n22221') == 99881)
24 assert(encuentra_maximo('32651\n71674\n42885\n22742\n72223') == 72871)
25 assert(encuentra_maximo('32659\n79674\n42886\n22742\n22229') == 97821)
26 assert(encuentra_maximo('32639\n79674\n42886\n22742\n22229') == 94839)
27
28 #Soluciones con m < n
29 assert(encuentra_maximo('41622\n42884\n42884\n27445\n42820') == 7241)
30 assert(encuentra_maximo('42622\n42684\n42684\n27445\n42220') == 6607)
31 assert(encuentra_maximo('44444\n44444\n44944\n44444\n44444') == 449)
32 assert(encuentra_maximo('44444\n44444\n44944\n44444') == 449)
33
```

Figura 3: Captura de pantalla de los tests propuestos para la resolución de la primera práctica obligatoria de la asignatura. Los tests probaban funcionalidad a implementar en el enunciado de la práctica en orden creciente de complejidad de manera que los alumnos pudieron, con anterioridad a la recepción de las calificaciones, conocer la calidad de sus soluciones en función del número de tests que se ejecutasen correctamente.

de duración utilizando un cuaderno similar a los otros empleados en las tareas y prácticas obligatorias.

6. Resultados

En esta sección se presentan los principales resultados de la evaluación de la metodología. Primeramente, Se realizó un estudio de la participación y asistencia a los talleres, así como de la entrega de tareas y prácticas obligatorias, con el objetivo de medir la tasa de abandono por parte del alumnado. El número de asistentes a la primera sesión (número máximo alcanzado) fue de 35 alumnos (81 %), número más o menos constante a lo largo del curso, aunque decreció hacia el final del mismo. Hecho llamativo es que 8 alumnos (19 %) nunca acudieron a ninguna de estas sesiones, ni entregaron ninguna de las tareas propuestas.

En lo referente a las entregas de ejercicios prácticos, la Figura 4 muestra la evolución del número de entregas y aprobados. Como ocurre en otras materias, se produce un decremento de la participación en la parte final del curso, cuando la acumulación de tareas es mayor y los estudiantes no cuentan con tiempo disponible para finalizar las tareas. A pesar de ello, se mantuvieron unos porcentajes altos de asistencia a las sesiones presenciales (sobre todo si se pondera con el número de asistentes a las sesiones al comienzo del curso), así como también de aprobado de las prácticas obligatorias. Como parte final del proyecto, se llevó a cabo una encuesta en una de las sesiones finales a la que contestaron 22 alumnos (51 % del total). El formulario se distribuyó en papel y consistía de una serie de bloques de preguntas relacionados con diferentes marcadores de calidad [4], la mayoría a responder empleando una

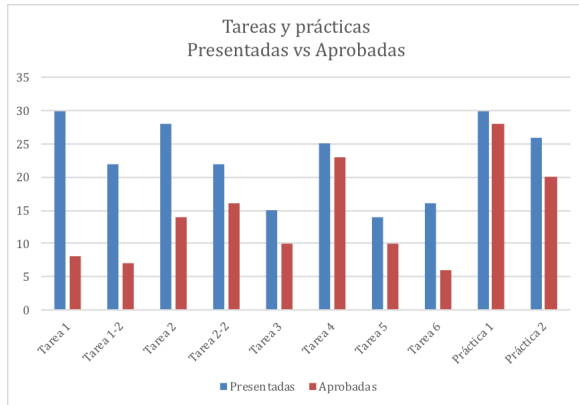


Figura 4: Evolución del número de alumnos que entregó las tareas y prácticas comparado con el número de aprobados. La participación bajó sensiblemente en aquellas últimas, quizás debido al hecho de que se solapaban con la entrega de las prácticas obligatorias.

escala de Likert. El objetivo del primer bloque de la encuesta era *comprender la percepción*, en relación a su utilidad, que tenía el alumnado sobre el *contenido de la asignatura*.

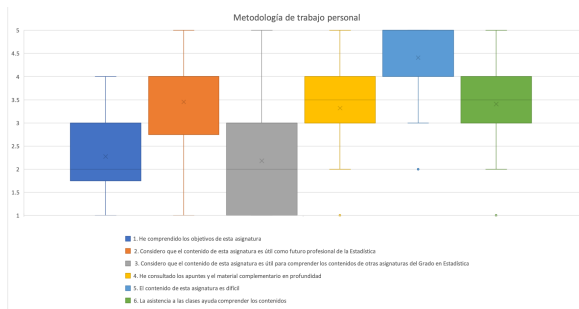


Figura 5: Visualización del primer bloque temático del cuestionario de satisfacción. La mayoría de alumnos percibieron la asignatura como difícil y valoraron la asistencia a clase de utilidad para comprender los contenidos del curso.

En la Figura 5 se muestran los resultados de este primer bloque, de los que se extraen las siguientes conclusiones:

- Los alumnos encuestados afirmaron no haber comprendido completamente los contenidos de la asignatura.
- En líneas generales confiaban en que el contenido de la asignatura les sería útil como futuros profesionales de la Estadística, pero no así para comprender los contenidos de otras asignaturas del grado.
- Los encuestados coincidieron en haber consultado los recursos disponibles y en que la asistencia a

las sesiones prácticas les fue útil para comprender los contenidos.

- Por último, mayoritariamente consideraron la asignatura como bastante difícil.

El siguiente bloque trataba de conocer *cómo los alumnos emplearon los cuadernos interactivos* durante el curso. Los resultados se muestran en la Figura 6 de la que se extrae que:

- La mayoría de alumnos emplearon los cuadernos para estudiar los contenidos del curso.
- Sin embargo, pocos los usaron para tomar apuntes en clase lo que habría supuesto una sustitución completa del formato papel. Especulando podríamos quizás alegrar esto a la falta de soltura con el manejo de los cuadernos o el lenguaje *Markdown*, aunque sería necesario ahondar de manera más rigurosa en esta cuestión para conocer las razones específicas de este hecho.
- Un alumno de los encuestados afirmó haber usado los cuadernos para tareas de programación no relacionadas directamente con la asignatura. Este hecho es interesante ya que da indicios de que los cuadernos podrían tener el efecto positivo de fomentar la programación creativa como se extrae de estudios de otros autores [5, 7].

En el tercer bloque se trataba de medir la *percepción del alumnado sobre las innovaciones introducidas* en la tecnología en referencia a la calidad de acceso al servicio, percepción general de la utilidad de la metodología y otros parámetros (Ver Figura 7). De acuerdo a los resultados, se pudo concluir que:

- La tendencia vista en otros bloques relacionada con la dificultad de la asignatura se replica aquí también. Los alumnos presentan dudas sobre la idoneidad de las prácticas semanales y el tiempo disponible para realizarlas.
- No queda claro si la metodología de aprendizaje se percibe como útil para comprender mejor el

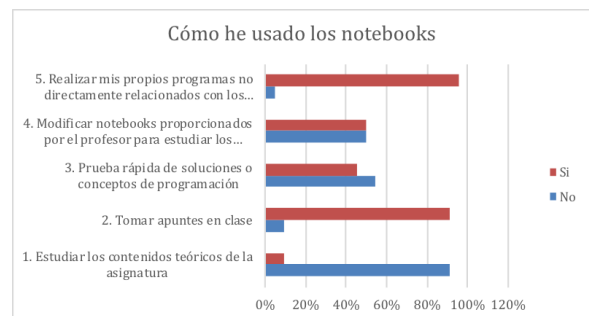


Figura 6: Distribución de resultados referente al uso de los cuadernos. La mayoría de usuarios los emplearon para estudiar los contenidos teóricos del curso.

contenido o lograr los objetivos del aprendizaje.

- El nivel de incidencias de acceso al servicio está dentro de límites razonables.
- Una gran mayoría coincide en que los cuadernos son una buena manera de aprender a programar, aunque si nos remitimos a lo visto en la encuesta inicial no podrían compararlo con otros métodos debido a su inexperiencia en el tema. Sería necesario investigar las causas por las cuales perciben positivamente el sistema.
- Finalmente, el trabajo por parejas fue satisfactorio para la mayoría.

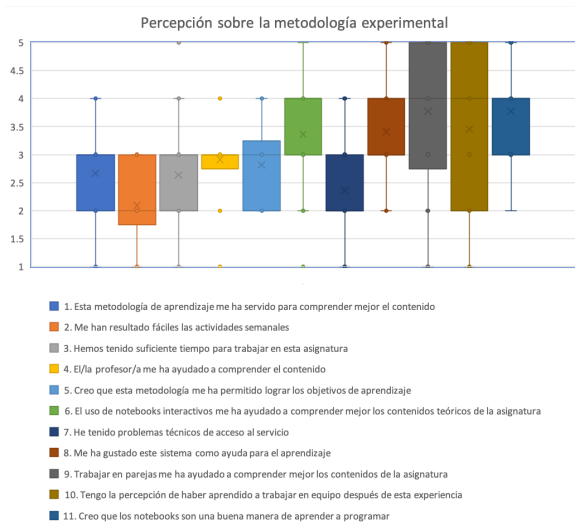


Figura 7: Resultados referentes a cómo los alumnos perciben la metodología experimental. La mayoría coincide en que la manera de trabajar propuesta había sido de su agrado.

Los resultados del cuarto bloque (Figura 8) muestran un nivel de *satisfacción* no muy alto con la asignatura, quizás debido a la alta dificultad percibida. Sin embargo parece haber un consenso en las otras dos preguntas: La mayoría de alumnos opinan que hubiesen aprendido menos estudiando por su cuenta los contenidos y, en menor medida, creen también que el enfoque empleado podría ser exportado para la docencia de otras asignaturas del grado. También se preguntó a los alumnos sobre la *utilidad percibida de otros elementos del ecosistema* de aprendizaje, como el campus virtual o las tutorías virtuales por correo electrónico y presenciales (Figura 9). A grandes rasgos, todos los recursos resultaron útiles salvo las lecturas recomendadas, aunque desconocemos los hechos que motivaron esta circunstancia. En siguientes revisiones de este método será necesario hacer un seguimiento más profuso del acceso a estas fuentes y su comprensión por parte de los alumnos.

En otro bloque temático de preguntas, los alumnos

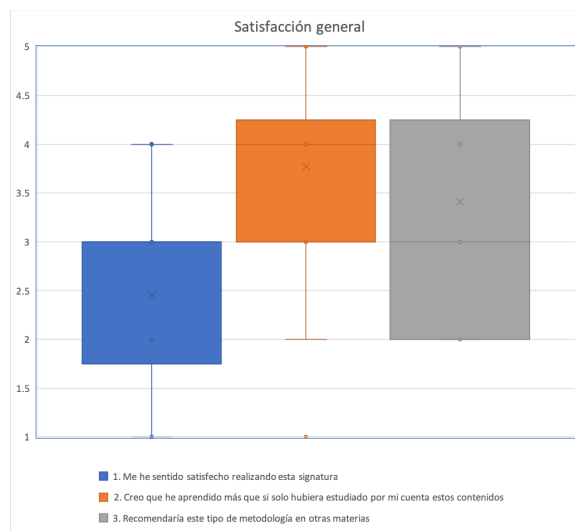


Figura 8: Satisfacción general de los alumnos con la asignatura.

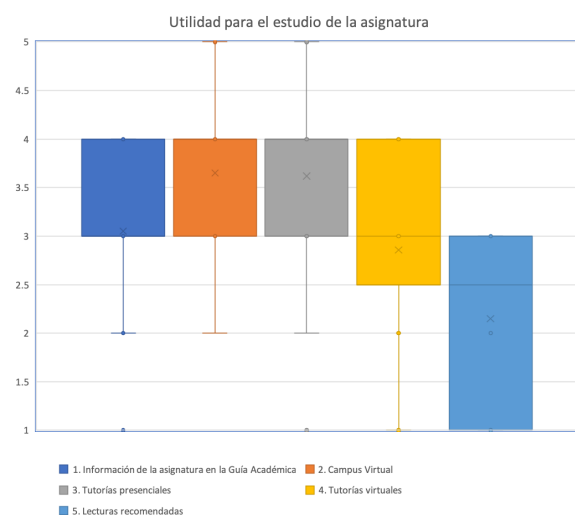


Figura 9: Utilidad de otros elementos, diferentes al cuaderno interactivo, para el estudio de la asignatura.

valoraron los *distintos métodos de aprendizaje*, situando en el lugar más alto de utilidad el aprendizaje mediante el trabajo en grupo, en el que se incluye el trabajo en parejas. Este marcador confirma que la técnica de programación por pares escogida fue adecuada y resultó en una experiencia beneficiosa y agradable para los participantes.

7. Conclusiones

En esta contribución se demostró cómo poner en práctica una arquitectura software en un curso de iniciación a la programación en lenguaje Python. Median-

te el uso del software adecuado, fue posible eliminar tareas redundantes que se identificaron como un obstáculo al aprendizaje en un curso de este tipo. Se vio también cómo el uso de cuadernos interactivos y la elección del lenguaje de programación fueron las adecuadas para que los alumnos asimilasen un contenido que es percibido como de gran dificultad. Esperamos en el futuro poder ahondar en algunas de las cuestiones planteadas a lo largo de esta memoria, como por ejemplo la implementación de ejercicios personalizados que se adaptan de manera más inteligente al progreso del alumno. En esta ocasión se contaba con un repositorio de ejercicios relativamente pequeño por haber sido esta la primera experiencia dentro del grupo empleando estas tecnologías. Finalmente, es importante reseñar que la metodología no fue capaz de reducir la dificultad de los contenidos percibida por los participantes. Como se pudo constatar durante el curso, y a pesar de la supuesta solvencia matemática del grupo, éste tuvo muchos problemas para manejar algunos de los conceptos de este tipo empleados en los ejercicios. Por lo tanto, es necesario realizar estudios comparativos más complejos en futuras experiencias docentes para localizar el origen exacto de esta dificultad.

8. Agradecimientos

Los autores quieren agradecer el apoyo recibido por parte del Vicerrectorado de Docencia y el Instituto de Biología Funcional y Genómica de la USAL para la realización de este proyecto de investigación.

Referencias

- [1] Jeff Brown. Using JupyterHub in the Classroom: Setup and Lessons Learned. *International Journal of Software Engineering & Applications (IJ-SEA)*, 9(2), March 2018.
- [2] Juan Cruz-Benito, Oriol Borrás-Gené, Francisco J García-Peñalvo, Ángel Fidalgo Blanco, and Roberto Therón. Extending MOOC ecosystems using web services and software architectures. In *Proceedings of the XVI International Conference on Human Computer Interaction*, page 52. ACM, 2015.
- [3] Stephen H Edwards. Using test-driven development in the classroom: Providing students with automatic, concrete feedback on performance. In *Proceedings of the International Conference on Education and Information Systems: Technologies and Applications EISTA*, volume 3. Citeseer, 2003.
- [4] Alicia García-Holgado, Francisco José García-Peñalvo, Juan José Mena Marcos, and Carina S González González. Inclusión de la perspectiva de género en la asignatura de Ingeniería de Software I. In *MID. Memorias de Innovación Docente de La Universidad de Salamanca*. 2017.
- [5] CJ García Monsálvez. Python como primer lenguaje de programación textual en la Enseñanza Secundaria. *Education in the Knowledge Society*, 18(2), 2017.
- [6] Jessica B. Hamrick. Creating and Grading IPython/Jupyter Notebook Assignments with NbGrader. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education, SIGCSE '16*, pages 242–242, New York, NY, USA, 2016. ACM.
- [7] Faraón Llorens Largo, Francisco J García-Peñalvo, Xavier Molero Prieto, and Eduardo Vendrell Vidal. La enseñanza de la informática, la programación y el pensamiento computacional en los estudios preuniversitarios. In *Education in the Knowledge Society (EKS)*, volume 18, pages 7–17. Ediciones Universidad de Salamanca, 2017.
- [8] Charlie McDowell, Linda Werner, Heather Bullock, and Julian Fernald. The Effects of Pair-programming on Performance in an Introductory Programming Course. In *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education, SIGCSE '02*, pages 38–42, New York, NY, USA, 2002. ACM.
- [9] Keith O'Hara, Douglas Blank, and James Marshall. Computational Notebooks for AI Education. In *The Twenty-Eighth International Flairs Conference*, April 2015.
- [10] Rodrigo Santamaría Vicente. Wikipedia como herramienta de aprendizaje en Sistemas Distribuidos (Ingeniería en Informática). In *MID. Memorias de Innovación Docente de La Universidad de Salamanca*. 2013.
- [11] Roberto Therón, Rodrigo Santamaría, and Felicidad García Sánchez. Interfaces de usuario imaginadas: El camino de ida y vuelta entre la ciencia y la ficción. In *MID. Memorias de Innovación Docente de La Universidad de Salamanca*. 2017.
- [12] José A. Troyano, Fermín Cruz, Mariano González, Carlos G. Vallejo, and Miguel Toro. Introducción a la Programación con Python, Computación Interactiva y Aprendizaje Significativo. *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática*, 3(0):223–230, June 2018.