

Proyectos de programación grupal en Python: ABP en el redictado del ciclo inicial universitario

Natalia Colussi¹ y Natalia Monjelat²

¹ Facultad de Ciencias Exactas, Ingeniería y agrimensura (UNR), Rosario, Argentina

² Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE, CONICET-UNR),
Rosario, Argentina
colussi@fceia.unr.edu.ar, monjelat@irice-conicet.gov.ar

Resumen. Este artículo analiza la aplicación de la metodología de Aprendizaje Basada en Proyectos y Problemas (ABP) en el desarrollo de un proyecto grupal de programación usando el lenguaje Python en la cátedra del redictado de Programación II para la Licenciatura Ciencias de la Computación. Los redictados en primer año recuperan y retienen a los estudiantes con dificultades para asimilar conceptos específicos del área y para sobrellevar los procesos tradicionales de evaluación universitarios. Se planteó un abordaje integral de la problemática aplicando un ABP guiado y supervisado por el equipo docente, con objetivos y controles semanales. Empleando el lenguaje Python junto al módulo *streamlit* para generar automáticamente un *front-end*, los grupos desarrollaron las funcionalidades que responden a las consultas solicitadas (*back-end*), mostrando ventajas y desventajas de las primeras estructuras de datos y conceptos aprendidos, ejercitándose sobre requerimientos reales. El análisis de las encuestas a los estudiantes revela cómo los saberes son puestos en práctica y cómo se logra apreciar la importancia del uso de metodologías de construcción de los programas al abordar un problema real de mayor magnitud. También se distingue cómo se logra superar mediante la estrategia de indagación, las dificultades sobre aquellos aprendizajes que deben darse de forma autónoma.

Palabras claves: Aprendizaje Basado en Proyectos y Problemas, Primer curso de Programación en Python, Didáctica en Ciencias de la Computación.

1 Introducción

En el presente artículo se analiza una experiencia llevada a cabo en la Facultad de Ciencias Exactas, Ingeniería y Agrimensura (FCEIA), de la Universidad Nacional de Rosario (UNR) para el aula del redictado de la materia Programación II, segunda materia disciplinar de la Licenciatura en Ciencias de la Computación (LCC), correspondiente al primer año de la carrera. En esta cátedra, se avanza sobre conceptos intermedios de la programación, desde el paradigma imperativo, empleando los lenguajes Python y C. Previamente los estudiantes ya cuentan con experiencia disciplinar en el área, tras haber cursado y aprobado la materia Programación I, donde mediante la programación funcional, aprenden los conceptos básicos de la programación, aplicando estrategias algorítmicas de resolución de problemas para modularizar sus programas, usando en este caso, el lenguaje Racket [10].

Considerando lo expuesto, y dada la esencia del redictado como política universitaria para la retención de estudiantes, se diseñó para esta instancia de la materia, una propuesta diferente a la transitada en el primer dictado, apoyándonos en el Aprendizaje Basado en Problemas (ABP), y en esta línea didáctica, particularmente, en el artículo de Frydenberg y Mentzer [9] donde los autores destacan los beneficios del ABP+Python para : el desarrollo del pensamiento crítico, la creatividad, la aprehensión de saberes, y la colaboración entre los pares. Como pilares, se tomaron los principios del pensamiento computacional [3], las estrategias de resolución de Polya [4] adaptadas por Thompson [5] para la programación, la estrategia pedagógica de Indagación [6] para apuntalar el aprendizaje autónomo, y la adquisición de habilidades profesionales disciplinares, llamadas *soft-skills* [7].

Desde estos aportes, se pensó e implementó la propuesta pedagógica del redictado de Programación II, la cual consistió en la realización de un proyecto de programación grupal en Python, combinando módulos específicos para automatizar la generación de un *front-end* como Streamlit¹, y módulos para procesar y graficar los datos, como *pandas*, *matplotlib*, etc. Para la puesta en marcha del proyecto, los estudiantes en grupos, recolectaron datos reales del repositorio público de AirBnb, eligiendo una ciudad dentro de este espacio para posteriormente generar preguntas que permitieran explorar: las características de los alojamientos en dicha ciudad, la dispersión en los barrios, la variedad en confort de las locaciones, etc. Luego se comenzó a trabajar sobre los aspectos de programación específicos para resolver cada proyecto. Durante todo el proceso, se evaluaron contenidos propios de la asignatura como: a) el uso de estructuras de datos simples: listas, tuplas, y diccionarios; b) el diseño modularizado del problema; c) la aplicación de la metodología específica de construcción de programas denominada: “La Receta Infalible” para programar [10]; y d) el desarrollo de *soft-skills*. En este trabajo, nos interesa conocer la opinión de los estudiantes frente a esta propuesta novedosa, ya que sus comentarios permiten avanzar en la revisión del diseño pedagógico del cursado.

2 Metodología

Para conocer las opiniones de los participantes sobre la propuesta, se diseñó y administró una encuesta a los estudiantes del cursado con preguntas abiertas y cerradas. Puntualmente para este trabajo se presentará el análisis de las siguientes preguntas:

1. *¿Cómo reconocieron qué estructuras de datos era mejor usar para modelar los datos dentro del programa?*
2. *¿Cómo vivenciaron el manejo de las librerías?*
3. *¿Qué fue lo que les permitió apreciar y/o distinguir que “la receta” para la construcción de programas los ayudó a construir “un mejor programa”?*

La misma se administró al finalizar el cursado de la asignatura y fue respondida por 22 alumnos. Para el análisis de las respuestas se trabajó con la identificación de patrones emergentes en los datos, desde una mirada cualitativa que permita relevar las voces de los participantes. Los resultados de dicho análisis se presentan en la sección siguiente.

¹ <https://streamlit.io/>

3 Resultados

En relación con la pregunta 1, vinculada al reconocimiento de cuál era la mejor estructura para modelar los datos, las respuestas se orientan hacia diferentes aspectos. Por un lado, se observa la utilización de tutoriales o documentación en línea como ayuda, mientras que otros participantes emplearon el método “prueba y error”, hasta dar con la estructura que mejor se adaptaba a lo que necesitaban. Asimismo, hay respuestas que explican con mayor nivel de detalle el proceso desarrollado: *“Nos pareció que el diccionario era lo que necesitábamos para crear nuestra función de filtrado gracias a su sistema de acceso a la información mediante claves”*.

Otras hacen referencia al uso de *dataframes*, en la simplificación del proceso: *“leyendo la documentación de pandas, nos dimos cuenta que los dataframes nos ofrecían una forma rápida y relativamente sencilla de trabajar con una cantidad masiva de datos, lo que nos ahorró mucho tiempo de producción y nos facilitó la construcción del programa”*. Finalmente, también se observan respuestas que señalan la importancia de buscar la manera más eficiente de recorrer los datos, tal como se observa en el siguiente extracto: *“nos dimos cuenta de que si usábamos cierta forma de representar los datos, habría que estar constantemente recorriendo el archivo así que utilizamos la que más eficiente era, (recorriendo 1 sola vez el archivo)”*.

Por otro lado, en cuanto al manejo de las librerías, se destaca en la mayoría de las respuestas, una dificultad inicial para comprender su funcionamiento, empleando adjetivos como “complicado” o “engorroso”. En algunos casos, se señalan puntualmente cuáles fueron las que más les costó usar: *“plotly.graph_obj fue la que más me costó entender personalmente”*, *“Complicado al principio, sobre todo para entender qué es lo que hacían aquellos comandos que queríamos usar, por ejemplo `st.map()` nos hizo renegar bastante. Después ya uno se familiariza.”*. Sin embargo, en todos los casos, se señala cómo luego su uso contribuyó al proceso, sobre todo destacando la posibilidad de acceso a la documentación y a las comunidades en los foros. En ese sentido los estudiantes indican: *“Una vez que nos adentramos en el funcionamiento de las librerías utilizadas, entendimos cómo estas herramientas facilitan el desarrollo de aplicaciones bastante sofisticadas”*, *“Al principio fue algo difícil ya que pensamos, que tanto `matplotlib` como `pandas`, pueden llegar a tener funciones complejas, como también sencillas, y que se debe tener paciencia para investigar y entender el funcionamiento de cada una, para luego modificar los datos y encarar una posible solución a un problema dentro del código.”*

Finalmente, en cuanto a la relación entre la receta y un buen programa, el análisis de las respuestas muestra aspectos tales como “orden del programa”, “código más limpio, estructurado”, “visualización clara de soluciones”. Por otro lado, se observan respuestas que señalan la modularización y la comprensión de funciones como aspectos a destacar de la receta. Asimismo, un punto que sobresale en el análisis es el hecho de facilitar la comprensión para todos los integrantes del grupo e incluso para quien pueda tener acceso al código en instancias posteriores: *“Al momento de que uno tenía que seguir escribiendo código por su parte, resultó ser indispensable la manera de estructurar el programa, realizar comentarios y demás para lograr una buena organización”*.

4 Conclusiones

Los resultados muestran que los estudiantes logran evaluar, aprender la mecánica y el estudio crítico necesario, a la hora de elegir qué estructura de datos es conveniente aplicar para resolver un problema real, apreciando la importancia de ésta etapa en la resolución del problema. Por otro lado, la conexión con el mundo laboral, instala la necesidad del “aprendizaje continuo”, la necesidad de disponer de tiempo para aprender algo nuevo y aplicarlo. También cobra relevancia emplear una metodología de trabajo y diseño de programas, cuando el desarrollo del software debe producirse a partir del trabajo de muchas personas, que interactúan e intervienen en la producción del código. La experiencia de implantar en la cátedra el ABP con Python resulta entonces muy satisfactoria, tanto para los estudiantes como para los docentes, marca un trayecto de aprendizaje genuino de los alumnos, quienes aprenden desde: la experiencia de resolver un problema desafiante, la comunicación y construcción de una solución con el otro, y el empleo de un proceso diferente de evaluación, aprobación.

References

- 168904240. Sánchez, P., Blanco, C.: Implantación de una metodología de aprendizaje basada en proyectos para una asignatura de Ingeniería del Software. En *Actas XVIII JENUI 2012*, Ciudad Real, Universidad Nacional de Cantabria, España (2012).
- 168904241. García Martín, J., Pérez Martínez, J.: Aprendizaje basado en proyectos: método para el diseño de actividades. *Revista Tecnología, Ciencia y Educación*, 5, 37-63 (2018).
- 168904242. Martínez Lopez, P.: Las Bases de la Programación. Publicado electrónicamente por la Universidad Virtual de Quilmes, La Plata, (2013).
- 168904243. Wing, J.M.: Computational thinking. *Commun. ACM*, 49(3), 33-35, (2006).
- 168904244. Polya, G.: *How To Solve It: A New Aspect of Mathematical Method*. Princeton: University Press, (1973).
- 168904245. Thompson S.: *Where do I begin? A problem solving approach in teaching functional programming*. LNCS (vol 1292). Springer, Berlin, Heidelberg, (1997).
- 168904246. Dostál, J.: *Inquiry-based instruction: Concept, essence, importance and contribution*. PhD thesis, Palacký University, Olomouc, Czech Republic, 2015.
- 168904247. Keogh S., Bradnum J., Anderson E. Improving professionalism in first year computer science students: Teaching what can't be taught. En *Proceedings of the 3rd Conference on Computing Education Practice (CEP '19)*, (2019). 1-4.
- 168904248. Frydenberg, Mark, and Kevin Mentzer. “From Engagement to Empowerment: Project-Based Learning in Python Coding Courses.” *EDISG Conference, Information Systems & Computing Academic Professionals*, 2020.
- 168904249. Felleisen, M., Findler, R., Flatt, M., Krishnamurthi, S. (2001). *How to Design Programs: An Introduction to Programming and Computing*. MIT Press, USA