

ABP+DataSet: Proyectos Grupales de Programación en Primer Año de la Universidad

Natalia Coluss¹[0009-0004-7146-5160] y Natalia Monjelat²[0000-0002-5043-8989]

¹ Facultad de Ciencias Exactas, Ingeniería y Agrimensura (FCEIA). Universidad Nacional de Rosario (UNR)

² Instituto Rosario de Investigaciones en Ciencias de la Educación (IRICE. CONICET-UNR)

{colussi@fceia.unr.edu.ar; monjelat@irice-conicet.gov.ar}

Resumen

Este artículo analiza una propuesta didáctica basada en la metodología de Aprendizaje Basado en Proyectos y Problemas (ABP), implementada en el redictado de la asignatura Programación II de la Licenciatura en Ciencias de la Computación, con el objetivo de conocer las opiniones de los participantes sobre la experiencia. En este caso, se propuso un trabajo grupal en el que se utilizaron Datasets con características específicas para practicar conceptos fundamentales como strings, tuplas, listas, diccionarios y archivos. El enfoque se centró en el manejo de grandes volúmenes de datos, cuya complejidad exige soluciones computacionales, y se empleó el lenguaje Python junto al módulo *streamlit* para el desarrollo de aplicaciones con front-end y back-end integrados. Al finalizar el curso, se realizó una encuesta a 35 estudiantes. Las respuestas reflejaron una valoración positiva: los Datasets fueron percibidos como novedosos, simples y relevantes por su conexión con datos reales. Sin embargo, también se identificaron dificultades, especialmente en la lectura de archivos CSV, la extracción de datos y el tratamiento de valores vacíos. En conclusión, la propuesta resultó innovadora y útil, fortaleciendo tanto el aprendizaje técnico como el trabajo colaborativo en la resolución de problemas.

Palabras clave: Aprendizaje Basado en Proyectos y Problemas, Primer curso de Programación en Python, Didáctica en Ciencias de la Computación, DataSet, Pensamiento Computacional.

PBL+DataSet: Group Programming Projects in the First Year of University

Abstract

This article analyzes a didactic proposal based on the Project and Problem-Based Learning (PBL) methodology, implemented in the second dictation of Programming II

within the Bachelor's Degree in Computer Science, aiming to acknowledge students' opinions about the experience. In this case, a guided group project was proposed, using specifically designed Datasets to practice fundamental concepts such as strings, tuples, lists, dictionaries, and files. The focus was on working with large volumes of data, which require computational solutions. Python was used along with the *streamlit* module to develop applications integrating both front-end and back-end components. At the end of the course, a survey was conducted with 35 students to gather their feedback. The responses reflected a positive perception: the Datasets were seen as novel, simple, and relevant due to their connection with real-world data. However, some difficulties were also reported, particularly with reading CSV files, extracting data, and handling missing values. In conclusion, the proposal proved to be innovative and beneficial, strengthening both technical skills and collaborative problem-solving abilities.

Keywords: Project And Problem Based Learning, DataSets, Python, First course of programming, Computational Thinking.

1 Introducción y contextualización

En este artículo se analiza una experiencia realizada en la Licenciatura en Ciencias de la Computación (LCC) en la Facultad de Ciencias Exactas, Ingeniería y Agrimensura (FCEIA), de la Universidad Nacional de Rosario (UNR), puntualmente en la cátedra del redictado de Programación II, de primer año.

En diferentes trabajos previos se ha señalado que la enseñanza de la programación en el ciclo inicial universitario presenta diversos desafíos, particularmente en los últimos años, ya que las carreras de informática han incrementado sus inscripciones y las cátedras de primer año se han superpoblado de alumnos, con perfiles diversos en cuanto a saberes y conocimientos previos (Colussi y Monjelat, 2023a;2023b; Colussi y Viale, 2024). Sin embargo, los recursos docentes para dictar las materias disciplinares en las carreras informáticas son escasos y los espacios áulicos de laboratorios con disponibilidad tecnológica actualizada no se han incrementado al ritmo de la matrícula. Todos estos factores contribuyen en diferente medida a que una gran mayoría de los estudiantes no logre aprobar las materias, por lo que la UNR ha implementado la posibilidad de un segundo dictado (redictado) para las materias de primer año en las distintas carreras, como medida de retención de los estudiantes en el ámbito universitario.

La experiencia en el redictado de Programación I de la LCC (Colussi y Viale, 2019; Colussi, Viale y Monjelat, 2021; Colussi, Monjelat y Viale, 2022; Colussi y Monjelat, 2022; Monjelat, Colussi y Viale, 2021) sirvió como insumo para diseñar la estrategia didáctica de la cátedra del redictado de Programación II. Asimismo, se tomaron los aportes del ABP (García Martín y Perez Martínez, 2018; Sánchez y Blanco, 2012) particularmente para el trabajo con Python (Frydenberg y Mentzer, 2020), el enfoque de resolución de problemas de Polya (1973) adaptado al diseño de programas por Thompson (1997), el enfoque del Pensamiento Computacional de Wing (2006), la estrategia pedagógica de Indagación (Dostál, 2015) y la adquisición

de las llamadas soft-skills mediante la incorporación de saberes profesionales tempranamente (Keogh et al., 2019).

La presente experiencia se enmarca en el Proyecto de Promoción Científico Tecnológica (PPCT-UNR 2024-2027, NRO: 80120230300052UR) aprobado por la Universidad Nacional de Rosario, radicado en la Facultad de Ciencias Exactas, Ingeniería y Agrimensura, titulado: “Aprendizaje Basado en Proyectos y Problemas para Cátedras de Programación en el Ciclo Básico Universitario”, el cual resulta ser la continuación del proyecto bienal 2020-2021/22 PPCT: 800 201901 00255 UR – “Estrategias Didácticas para el Aprendizaje y la Enseñanza del Pensamiento Computacional en el Nivel Académico Universitario”. Ambos proyectos bajo la dirección de las autoras de este artículo.

2 Propuesta didáctica

El diseño e implementación de la propuesta didáctica del redictado de Programación II, consistió en la realización de un proyecto de programación grupal en Python, donde se desarrolló una aplicación web interactiva de visualización y consulta de información a partir de un Dataset, de una forma sencilla y práctica. Visualmente la página web tiene el aspecto de un tablero de información (*dashboard*) con entradas para realizar consultas sobre los datos. El *front-end* es logrado mediante la composición de métodos (*widgets*) específicos de entrada y salida, ofrecidos como funciones *built-in* del módulo *Streamlit*¹, sin tener que programar nada directamente en HTML. Todo el *back-end* de la aplicación web, se desarrolla en Python, importando en el programa principal el módulo Streamlit, junto a otras componentes, como por ejemplo, el módulo *Matplotlib*² - usado para graficar datos con distintas representaciones: gráficos de barras, de torta, etc., y el módulo *Pytest*³ - usado para realizar el testing unitario de las funciones desarrolladas.

Para la puesta en marcha del proyecto, los estudiantes se reúnen en grupos de tres integrantes y se les asigna un Dataset de repositorios públicos (datos de empresa de alocación de viviendas conocida como AirBnb⁴, datos sobre los combustibles y energías en Argentina⁵, espacios de entretenimiento y esparcimiento de la ciudad de Buenos Aires⁶, datos sobre los registros y las poblaciones aborígenes del país⁷, y por último datos sobre las escuelas, en todos los niveles⁸, de la provincia de Buenos Aires), y cada grupo procede a la elaboración de un conjunto de 6 preguntas a realizar sobre los datos. Algunas de las preguntas se caracterizan por ser

¹ <https://streamlit.io/>

² <https://matplotlib.org/>

³ <https://docs.pytest.org/en/stable/index.html>

⁴ <https://insideairbnb.com/get-the-data/>

⁵ <http://datos.energia.gob.ar/dataset/>

⁶ <https://data.buenosaires.gob.ar/dataset/espacios-culturales/>

⁷ <https://datos.jus.gob.ar/ro/dataset/listado-de-comunidades-indigenas/>

⁸ <https://catalogo.datos.gba.gob.ar/dataset/establecimientos-educativos/>

de análisis dinámico (van a modificarse según la entrada del usuario) y otras serán estáticas (no dependerán de la entrada de datos). Las salidas a posteriori del procesamiento de los datos deben incluir tablas, gráficos de barra o de torta, y ubicaciones en un mapa. Todo el manejo de E/S se realiza mediante los widgets que ofrece el módulo Streamlit y se programa desde Python incorporando el módulo mencionado e invocando la funcionalidad que se necesita. Luego de este proceso de generación de preguntas, más el control, revisión y depuración de las mismas por parte de los docentes asignados a la supervisión de cada grupo, se comienza a trabajar sobre los aspectos de programación específicos para resolver los problemas al interior de cada proyecto.

Semanalmente, se trabaja con objetivos que se asignan de manera general a todos los grupos, que se publican en un documento en el aula virtual, y todos los estudiantes acceden a ellos al iniciar la semana. Los docentes que supervisan los grupos verifican que los estudiantes puedan cumplirlos y avanzar hacia la concreción del proyecto. Se asignan un control semanal para cada grupo, con un lapso de tiempo de unos 15 minutos a cada grupo, y un total de 5 o 6 grupos por docente. Los controles se llevan a cabo en uno de los días de clase, durante el horario de laboratorio. Si algún grupo requiere más tiempo de consulta y/o control, entonces se lo cita a consulta para resolver los problemas que tengan. Una vez cumplido el plazo y junto a la entrega del proyecto, se realiza la exposición del mismo donde los estudiantes comparten los logros con el resto de la cátedra.

3 Metodología

Con el objetivo de conocer las opiniones de los participantes sobre la propuesta, se diseñó una encuesta que incluía preguntas abiertas y cerradas y que fue administrada al grupo de estudiantes que participó del redactado. Para este trabajo en particular, se presentará el análisis de las siguientes preguntas:

1. *¿Qué dificultades tuvieron durante el proceso de programación?*
2. *Los DataSet sobre los que trabajamos fueron extraídos de espacios públicos, municipales, provinciales, nacionales, e internacionales; son datos reales de nuestro país, nuestra cultura, economía, y usos y costumbres: ¿Qué apreciación te merece utilizar este tipo de DataSet? ¿Te resultaron interesantes? ¿Te gustaría trabajar con algún tipo de dato específico? ¿Lo habías hecho antes?*

La encuesta se administró al finalizar el cursado de la asignatura y fue respondida por 35 alumnos. Para el análisis de las respuestas se trabajó con la identificación de patrones emergentes en los datos, desde una mirada cualitativa que permite relevar las voces de los participantes. Los resultados de dicho análisis se presentan en la sección siguiente.

3 Resultados

En relación con la **pregunta 1**, vinculada a las dificultades en el proceso de programación, las respuestas se orientan hacia tres categorías diferentes:

- dificultades en la recuperación y manipulación de los datos del archivo,
- problemas en el uso y/o aplicación de librerías como *Streamlit* o *matplotlib*, y
- problemas de programación, diseño y manipulación de estructuras de datos.

En cuanto a la **primera categoría**, se observaron 10 respuestas que destacan dificultades relacionadas con el tratamiento de los datos de los archivos CSV, la separación de las filas, o el filtrado de celdas vacías o mal escritas, es decir, aquellos problemas que están relacionados con la manipulación de strings. Este tipo de complicaciones surgen porque muchas veces el manejo de este tipo de datos en la práctica se suele simplificar o no se valora la importancia de comprender bien su manipulación. Luego, en los problemas reales, ocurre que recuperar una línea de un archivo y procesar su contenido es concretamente manipular una string. Entonces aquí aparece la primera confrontación con la realidad y la necesidad de recuperar esos conocimientos vistos previamente. Algunas de las respuestas que manifiestan lo antes mencionado se pueden observar en la fig. 1.

"La mayor dificultad que tuvimos es que primeramente tuvimos un archivo csv con 2 separadores y nos era casi imposible poder leer los datos"(E14).

"Ninguna dificultad mayor, salvo algunos problemas de escritura del dataset (casillas vacías, casillas mal escritas, etc)"(E7).



"Hubo un pequeño problema en como se separaban las "filas" del archivo csv, pero lo resolvimos haciendo la función stringlist. Y hubo también un percance en las celdas vacías, pero conseguimos filtrarlas fácilmente."(E28)

"El csv tenía muchos datos vacíos o sucios que hacían más difícil la lectura con los códigos."(E22)



Fig. 1.: Extractos de las respuestas a la pregunta 1, categoría 1.

En relación con la **segunda categoría**, se recuperaron 14 respuestas que mencionaban diferentes aspectos vinculados al manejo de las librerías. Por un lado, se observa una dificultad relacionada con el acceso y la lectura de la documentación, lo cual a su vez forma parte de las acciones esperadas por parte del estudiantado para resolver la actividad propuesta, y por el otro lado aparecen los problemas que conlleva traducir un lenguaje de programación a aquello que queremos o necesitamos resolver computacionalmente. Éstos inconvenientes radican en la maduración disciplinar de la persona que programa. La mayoría de los estudiantes vienen de un proceso de aprendizaje incremental de la programación; muchos de ellos no tienen conocimiento previo sobre programación y aprenden de las materias disciplinares dentro de la carrera. Por ejemplo, en la primera cátedra de Programación I, se utiliza un lenguaje

funcional, Dr. Racket⁹, donde las consultas sobre las dudas del lenguaje y el debugging del programa se hacen dentro del mismo entorno de programación, el cual está especialmente diseñado y pensado didácticamente para programadores/as que se inician en la actividad. Por lo cual, es la primera vez que los estudiantes se enfrentan con la lectura de documentación pensada para una persona avanzada que programa, debiendo indagar al respecto, descifrar su contenido, practicar con la “prueba y error” de los ejemplos que se dan, usarla y adecuarla correctamente. Esto genera una segunda confrontación con la realidad del trabajo de la persona que programa, y resulta entonces natural que los/as estudiantes lo señalen como una dificultad. En la fig. 2 se muestra algunas de las respuestas más significativas al respecto de lo mencionado.

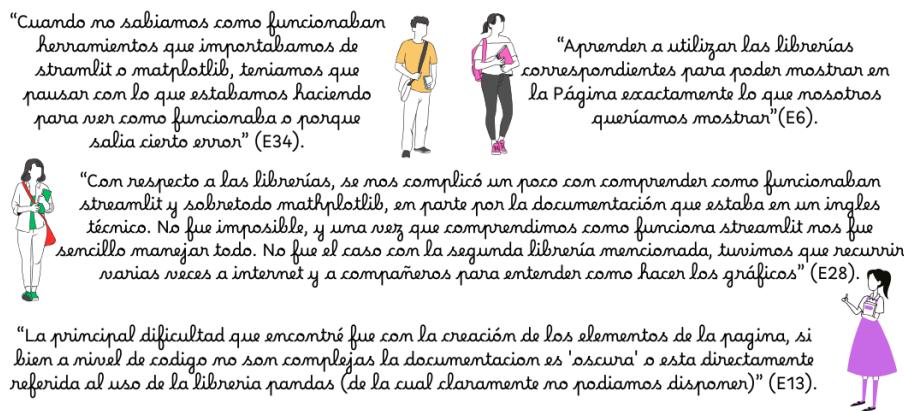


Fig. 2: Extractos de las respuestas a la pregunta 1, categoría 2.

Finalmente, se identificaron 12 respuestas en la **categoría 3** que reflejan problemas de programación, diseño y manipulación de las estructuras de datos que deben trabajarse con el proyecto, señalando también dificultades con la creación de los elementos. En este sentido, el trabajo de combinar diccionarios, tuplas y listas adecuadamente para utilizarlas en el proyecto, resulta un gran desafío. Estas estructuras son recientemente aprendidas y puestas en acción en el proyecto. Identificar cuál es una buena representación para el problema que deben resolver no es una tarea sencilla, sino que implica un trabajo complejo de abstracción que se realiza sobre los datos y las respuestas que buscan brindar.

Leer la información del Dataset, limpiar estos datos y abstraerlos en una o varias estructuras de datos es el 50% del trabajo del proyecto. A partir de ese momento, lo que queda es programar las funciones que los procesen para responder a las preguntas que el grupo planteó previamente sobre los datos. Esta etapa implica un trabajo de discusión dentro del grupo, con muchas propuestas, y requiere de una mirada crítica que se ejercita ampliamente, siendo uno de los objetivos que se pretende en la asignatura. Cabe aclarar que, como se mencionó previamente, cada etapa es

⁹ <https://racket-lang.org/>

supervisada por los docentes, desde una mirada de acompañamiento y guía del proceso de aprendizaje. La fig. 3 recoge voces directas que ilustran lo expuesto.

"Al principio nos costó pensar una función que una sola vez el archivo y que nos retorno la información que necesitábamos. También en pensar funciones que sea eficientes a la hora de leer el dataset, en el sentido de que no se colapse el replit." (E10)



"Una de las dificultades fue que no funcionaba el gráfico de barras, mostraba el gráfico pero vacío, pero al final nos dimos cuenta que la función que daba los datos de cada barra estaba devolviendo 0, y ahí pudimos solucionar el problema..." (E21)

"Las dificultades que tuvimos fueron trabajar con Streamlit, no conocíamos la herramienta y al principio se nos dificultó comenzar. Pero luego al buscar más y leer pudimos ir aclarando dudas. Otra dificultad fue crear una única estructura, que nos sirva para todos los archivos." (E27)

"En pensar una función la cual sería la estructura de todo el código, y también la parte donde compartimos funciones, con los diferentes preguntas." (E30)

Fig. 3: Extractos de las respuestas a la pregunta 1, categoría 3.

En cuanto a la **pregunta 2**, relacionada con el uso y manipulación de los Datasets, las respuestas se pueden agrupar en tres categorías también:

- experiencia y apreciación positiva al trabajo con Datasets,
- sugerencias de Datasets o temáticas de interés,
- aprendizajes relevados por los estudiantes en el proceso de concreción del proyecto.

Respecto a la **categoría 1**, se observaron 28 respuestas indicando una primera vez en el trabajo con Datasets, y éstas a su vez destacan que la experiencia fue positiva. Cómo lo muestran los extractos en la fig. 4, frases como “experiencia interesante”, “experiencia grata”, “fue interesante trabajar con Datasets”, “trabajar con algo importante y relevante”, “gran experiencia” ponen de manifiesto cómo cada estudiante vivió positivamente el proceso de concreción del proyecto de programación.

En lo que concierne a la **categoría 2**, se distinguen 10 respuestas de las cuales 5 señalan un interés por continuar trabajando con datos públicos de airbnb para el proyecto por lo interesante que éstos resultan y por lo que permiten consultar sobre ellos. Asimismo, otros estudiantes indicaron que les gustaría trabajar con datos de la provincia de Santa Fe, de la ciudad de Rosario, de economía y turismo, de salud mental y de comercio electrónico. Todas estas respuestas resultan de sumo interés para esta investigación puesto que permiten ampliar temáticas de interés al mismo tiempo que validan los datos que ya se utilizan.

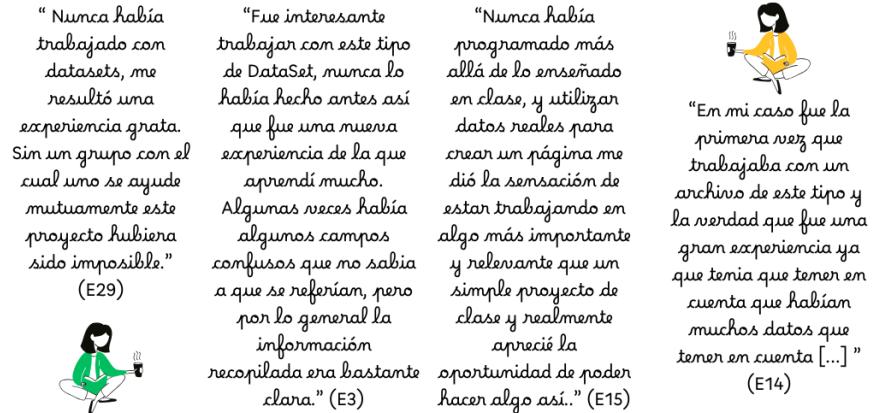


Fig. 4: Extractos de las respuestas a la pregunta 2, categoría 1.

Por último, en referencia a la **categoría 3**, 21 respuestas señalan saberes aprendidos durante la realización del proyecto que se agrupan en dos sub-categorías. La primera sub-categoría refiere a los aprendizajes vinculados a la programación, donde se pueden rescatar conocimientos asociados al pensamiento computacional, como trabajar con grandes volúmenes de datos y la importancia que adquiere la computación para su procesamiento, como el orden, la modularización y contar con una metodología de trabajo para abordar problemas complejos. Por su parte, la segunda sub-categoría refiere a aprendizajes ganados al procesar la información de los Dataset, en cuanto a los diferentes temas que los mismos abordaban. La fig. 5 señala algunos extractos de los dichos textuales de los estudiantes en relación a estos puntos.

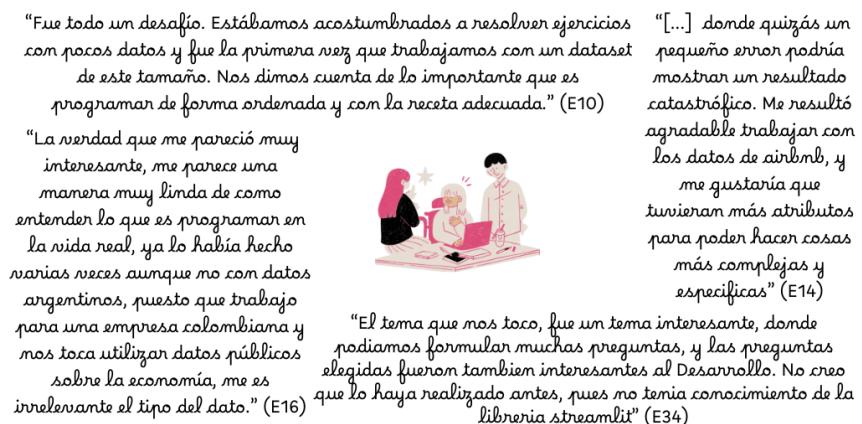


Fig. 5: Extractos de las respuestas a la pregunta 2, categoría 3.

Conclusión

Considerando el análisis presentado, es posible señalar que la implementación del Aprendizaje Basado en Proyectos y Problemas (ABP) en la cátedra de Programación II, utilizando Python y trabajando con Datasets reales, ha resultado ser una experiencia altamente enriquecedora tanto para estudiantes como para docentes. Los resultados muestran que la propuesta fue percibida como innovadora, interesante y significativa, favoreciendo un aprendizaje genuino. En este sentido, es posible señalar que del total de estudiantes que realizaron el proyecto de programación grupal ($n=53$), aprobaron el final de la materia un poco más del 55% ($n=29$). Y si consideramos la cantidad de estudiantes cursantes regulares que se presentaron a rendir en las primeras cuatro mesas de examen, el 82% aprobó el final. A su vez, éstos valores nos muestran que, sobre la cantidad de estudiantes que inician el curso y realizan al menos al menos un examen ($n=55$) el 96% sigue hasta el final el cursado, logrando regularizar la materia. Sostener todo el trabajo y lograr regularizar la materia es muy significativo, permite cursar las materias que son correlativas sin atrasarse en la carrera y predispone al estudiantado a continuar sus estudios y rendir los finales.

El trabajo con grandes volúmenes de datos, pese a presentar algunas dificultades iniciales relacionadas con la lectura de archivos CSV, la extracción de datos y la gestión de datos vacíos, fue valorado positivamente por su conexión con contextos reales y su relevancia en el mundo profesional. A través de esta metodología, los estudiantes no sólo reforzaron conocimientos técnicos en programación, sino que también desarrollaron habilidades fundamentales como la evaluación crítica de estructuras de datos adecuadas para distintos problemas, la comunicación efectiva, el trabajo en equipo y la construcción colaborativa de soluciones.

La experiencia evidenció además la importancia de integrar metodologías de diseño de programas centradas en la colaboración, preparando a los alumnos para los desafíos reales del desarrollo de software, donde la interacción y el consenso son esenciales. El trabajo grupal en programación no sólo permite a los estudiantes desarrollar habilidades técnicas, sino que también enriquece su experiencia individual y colectiva, fomentando la colaboración, el aprendizaje mutuo y la creación de soluciones más eficaces. Por lo dicho, este tipo de experiencias fomenta el desarrollo personal, a través de la comunicación efectiva, preparándolos para los desafíos del mundo real en el ámbito de la programación y el desarrollo del software. En este sentido, se logra conectar con el mundo laboral y la necesidad del “aprendizaje continuo”, teniendo en cuenta que el desarrollo del software debe producirse a partir del trabajo de muchas personas, que interactúan e intervienen en la producción del programa.

Considerando lo expuesto, se evidencia el potencial del ABP en el ámbito de la programación, por lo cual resulta necesario avanzar en el estudio de los procesos de enseñanza y aprendizaje en el contexto universitario desde este tipo de metodologías.

Referencias

- Colussi, N. y Viale, P. (2019). Actividades de Programación Grupales para Primer año de la Licenciatura en Ciencias de la Computación - Experiencias Didácticas en el Aula. Versión Extendida. En *Jornadas de CyT de la UNR, (JorCyT)*.
- Colussi, N., Viale, P. y Monjelat, N. (2021). Vidriera de proyectos: una modalidad de evaluación posible en tiempos de virtualidad. En *Actas de II WITE - TRANSFORMACIÓN DIGITAL. DESAFÍOS DE LA EDUCACIÓN SUPERIOR*, 1–10.
- Colussi, N., Monjelat, N. y Viale, P. (2022). El ABP en el redactado de materias de programación: una experiencia en la educación superior. En Actas de las Segundas Jornadas de Didáctica de las Ciencias de la Computación (JADiCC). Universidad Nacional del Nordeste.
- Colussi, N. y Monjelat, N. (2022) Estrategias didácticas para el aprendizaje y la enseñanza del pensamiento computacional en el nivel académico universitario. En Libro de actas - XXVIII Congreso Argentino de Ciencias de la Computación - CACIC 2022. Universidad Nacional de la Rioja.
- Colussi, N. y Monjelat, N., (2023a). Python + ABP: una experiencia innovadora en las aulas universitarias de programación. En *XV Jornadas Internacionales de Innovación Docente: Nuevos Ecosistemas para Nuevas Modalidades de Aprendizaje*, Universidad de Oviedo. https://inie.uniovi.es/c/document_library/get_file?uuid=c11bbce2-7303-4f3a-8cd7-d9767c9b4450&groupId=314473
- Colussi, N., & Monjelat, N. (2023b). Proyectos de programación grupal en Python: ABP en el redactado del ciclo inicial universitario. *Memorias De Las JAIIo*, 9(9), 61-64. <https://ojs.sadio.org.ar/index.php/JAIIo/article/view/700>
- Colussi, N., y Viale, P. (2024). Programación en Grupo usando Python+ABP: Una Experiencia Enriquecedora en el Aula Universitaria. En Actas de las IV Jornadas de Didáctica de las Ciencias de la Computación (JADiCC). Universidad Nacional de Río Cuarto.
- Dostál, J. (2015). Inquiry-based instruction: Concept, essence, importance and contribution. PhD thesis, Palacký University, Olomouc, Czech Republic.
- Frydenberg, M., y Mentzer, K. (2020) From Engagement to Empowerment: Project-Based Learning in Python Coding Courses. En EDISG Conference, Information Systems & Computing Academic Professionals.
- García Martín, J. y Pérez Martínez, J. (2018) Aprendizaje basado en proyectos: método para el diseño de actividades. *Revista Tecnología, Ciencia y Educación*, 5, 37-63.
- Keogh S., Bradnum J., y Anderson E. (2019). 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), 1-4.
- Monjelat, N.; Colussi, N. y Viale, P. (2021). Introducción a la Programación en carreras terciarias y universitarias de Computación: estado del arte en el contexto argentino. En *Jornadas de Ciencia y Técnica de la UNR*.
- Polya, G. (1973). *How To Solve It: A New Aspect of Mathematical Method*. Princeton: University Press.
- Thompson S. (1997). *Where do I begin? A problem solving approach in teaching functional programming*. LNCS (vol 1292). Springer, Berlin, Heidelberg.
- Sánchez, P. y Blanco, C. (2012). 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.