

Functional Programming in Academia, an Imminent Need

En el ámbito académico, la enseñanza de la programación funcional presenta desafíos particulares debido a sus principios conceptuales y a menudo abstractos. La falta de familiaridad con conceptos como funciones puras, inmutabilidad y recursión dificulta la comprensión de los estudiantes y limita su capacidad de aplicación de estos principios. Además, la programación funcional se percibe como menos relevante en comparación con paradigmas establecidos como la programación orientada a objetos, lo que disminuye el interés y la motivación en aprenderla. Por lo anterior, actualmente se requiere que en la academia se promueva el desarrollo de cursos que impacten el interés y la motivación de los estudiantes hacia la programación utilizando el paradigma funcional, mediante entornos de aprendizaje prácticos, presentación de casos de uso que promuevan la eficiencia, complejidad y mantenibilidad del código.

La pertinencia del estudio radica en la creciente importancia de la programación funcional en el panorama tecnológico y en la necesidad de proporcionar una guía concreta para escribir programas funcionales y, al mismo tiempo, utilizar eficazmente estos patrones en la academia. Con el evento de paradigmas de desarrollo de software mayormente orientados a la concurrencia y la eficiencia en la gestión de grandes conjuntos de datos, la comprensión de la programación funcional se vuelve crucial para los estudiantes del campo de la informática.

Un programador principiante necesita entender funciones, variables, constantes y condicionales. Asimismo, el uso de un lenguaje orientado a objetos implica comprender sobre clases, campos, métodos y herencia, además de los conceptos necesarios en un lenguaje funcional. En este último caso se introducen conceptos como la inmutabilidad, implica que los datos no pueden cambiar después de ser creados, lo que simplifica la comprensión del código y reduce los errores al eliminar efectos secundarios inesperados; las funciones puras, aquellas que producen el mismo resultado dado el mismo conjunto de entradas, sin efectos secundarios externos, lo que facilita la depuración y la comprensión del programa; las mónadas, estructuras que encapsulan operaciones con efectos secundarios, permitiendo gestionarlos de manera controlada y predecible en un entorno funcional, mientras que los patrones pueden definirse como soluciones comunes a problemas recurrentes, proporcionando un enfoque estandarizado para resolverlos, mejorar la legibilidad y mantenibilidad del código (Felleisen et al. 2004).

Dado lo anterior, y según expertos en el área como Wallingford (2002), la manera en la que se puede organizar el proceso de enseñanza de la programación funcional comprende:

Desarrollo de las clases: Exponer un problema mediante un patrón donde se expliquen las fuerzas que influyen en la solución.

Creación de asignaciones: Seleccionar problemas de tarea y exámenes que exploren cada patrón y fomenten a los estudiantes a elegir entre diferentes técnicas basadas en la solución del problema.

Evaluación del diseño: Utilizar un lenguaje de diseños de los estudiantes en función de cómo abordan las fuerzas del problema y cómo aplican los patrones de manera efectiva en la solución.

Luego de la aplicación de la metodología propuesta en un caso de estudio, se espera una comprensión sólida de los conceptos fundamentales de programación funcional, así como la capacidad de aplicar estos en la resolución de problemas prácticos, la habilidad para comparar y contrastar los paradigmas de programación funcional y orientada a objetos, el desarrollo de habilidades de resolución de problemas y la preparación para adaptarse a nuevos entornos.

Referencias

Castro-Romero, A., González-Sanabria, J.-S. (2021). Experiencias universitarias de aula en la introducción a la programación. Cuaderno de Pedagogía Universitaria, 18(35), 85-94. <https://doi.org/10.29197/cpu.v18i35.415>

Felleisen, M., Findler, R. B., Flatt, M., Krishnamurthi, S. (2004). The structure and interpretation of the computer science curriculum. Journal of Functional Programming, 14(4), 365-378. <https://doi.org/10.1017/s0956796804005076>

Khanfor, A., Yang, Y. (2017). An Overview of Practical Impacts of Functional Programming. En 24th Asia-Pacific Software Engineering Conference Workshops, 2017. <https://doi.org/10.1109/apsecw.2017.27>

Thompson, S., Wadler, P. (1993). Functional Programming in Education-Introduction. Journal of Functional Programming, 3(1), 3-4. <https://doi.org/10.1017/s0956796800000563>

Wallingford, E. (2002). Functional Programming Patterns and Their Role in Instruction. University of Iowa.
<https://cs.uni.edu/~wallingf/patterns/papers/fdpe2002/fdpe2002-long.pdf>

Wilson, G. (2022). Functional programming in education [Video].
<https://www.youtube.com/watch?v=nAdX5w-IHLY>