**PARADIGMAS DE PROGRAMACIÓN**

**¿Qué es un Paradigma de Programación?**

Un paradigma de programación es una manera o estilo de programación de software. Existen diferentes formas de diseñar un lenguaje de programación y varios modos de trabajar para obtener los resultados que necesitan los programadores.  Se trata de un conjunto de métodos sistemáticos aplicables en todos los niveles del diseño de programas para resolver problemas computacionales. (Canelo, 2020)

|  |  |  |  |
| --- | --- | --- | --- |
| **PARADIGMAS DE PROGRAMACIÓN** | | | |
| **PARADIGMA** | **DESCRIPCIÓN** | **EJEMPLOS** | **USO** |
| **Paradigma Imperativo** | Los programas consisten en una sucesión de instrucciones o conjunto de sentencias, como si el programador diera órdenes concretas. El desarrollador describe en el código paso por paso todo lo que hará su programa. | * Pascal * COBOL * FORTRAN * C * C++ | Uso frecuente a día de hoy en múltiples proyectos debido a su buena legibilidad y que es fácilmente comprensible, minimizando el riesgo durante cualquier tipo de edición. |
| **Paradigma Declarativo** | Este paradigma no necesita definir algoritmos, puesto que describe el problema en lugar de encontrar una solución al mismo. Este paradigma utiliza el principio del razonamiento lógico para responder a las preguntas o cuestiones consultadas. | * Prolog * Lisp * Scala * Java * Kotlin | Utilizados mayormente en casos en donde se requiera eficiencia y en donde se busque un mantenimiento independiente. |
| **Programación Orientado a Objetos** | En este modelo de paradigma se construyen modelos de objetos que representan elementos (objetos) del problema a resolver, que tienen características y funciones. Permite separar los diferentes componentes de un programa, simplificando así su creación, depuración y posteriores mejoras. La programación orientada a objetos disminuye los errores y promociona la reutilización del código. | * Java * Python * C# * ADA * C++ * Ruby * Eiffel * Objetive C | Utilizados cuando se busca una simplificación en su creación, facilitando con esto la depuración y posterior creación de mejoras. Es una gran alternativa debido a que además permite la reutilización del código. Su estilizo se acerca mucho a cómo expresaríamos las cosas en la vida real, facilitando su comprensión. |
| **Programación Reactiva** | La programación reactiva, o Reactive Programming, es un paradigma enfocado en el trabajo con flujos de datos finitos o infinitos de manera asíncrona, permitiendo que estos datos se propaguen generando cambios en la aplicación, es decir, “reaccionan” a los datos ejecutando una serie de eventos. | * Proyect Reactor * Rx Java * React/Angular | Utilizado en casos en donde se necesite operar y manejar streams (flujo de datos) de datos asíncronos. Se usa en sistemas en donde se necesite trabajar con datos que no se saben cuándo se generan, pero que se quiere reaccionar a ellos y actuar en consecuencia. |
| **Programación Funcional** | El nombre ya lo sugiere: la programación funcional o functional programming se centra en las funciones. En un programa funcional, todos los elementos pueden entenderse como funciones y el código puede ejecutarse mediante llamadas de función secuenciales. | * LISP * ML * Haskell * OCaml * F# * Erlang * Clojure | Usado frecuentemente por su facilidad de combinación con la programación imperativa y orientada a objetos. Además de que a día de hoy es una buena opción porque genera un código más preciso y corto, facilitando su verificación. |
| **Programación Orientada a Aspectos** | La programación orientada a aspectos (POA) es un paradigma de programación que basa su filosofía en tratar las obligaciones transversales de nuestros programas como módulos separados (aspectos) para lograr una correcta separación de responsabilidades. | * C++ * Java * Perl | Utilizado en entornos en donde se requiera un código fuente más “limpio”, además de que brinda la posibilidad de mezclarse con cualquier otro paradigma de programación, facilitando su interconexión. |

# Referencias

Canelo, M. M. (09 de Junio de 2020). *Profile - ¿Qué son los Paradigmas de Programación?* Obtenido de Profile.es: https://profile.es/blog/que-son-los-paradigmas-de-programacion/

García, J. M. (06 de Marzo de 2018). *GFT - Claves para entender la Programación Reactiva.* Obtenido de Blog GFT: https://blog.gft.com/es/2018/03/06/el-paradigma-de-la-programacion-reactiva-claves-para-entenderla/

IONOS, D. G. (11 de Febrero de 2020). *Digital Guide IONOS - Programación Funcional.* Obtenido de Digital Guide IONOS: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/programacion-funcional/

IONOS, D. G. (24 de Febrero de 2020). *Digital Guide IONOS - Programación Imperativa.* Obtenido de Digital Guide IONOS: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/programacion-imperativa/

Pulido, M. (23 de Septiembre de 2019). *Slash Mobility - Todo lo que deberías saber sobre Programación Reactiva.* Obtenido de Slash Mobility: https://slashmobility.com/blog/2019/09/programacion-reactiva/#:~:text=La%20programaci%C3%B3n%20reactiva%2C%20o%20Reactive,ejecutando%20una%20serie%20de%20eventos.

Ruelas, U. (18 de Mayo de 2017). *Coding or Not POA.* Obtenido de Coding or Not: https://codingornot.com/que-es-la-programacion-orientada-a-aspectos-aop